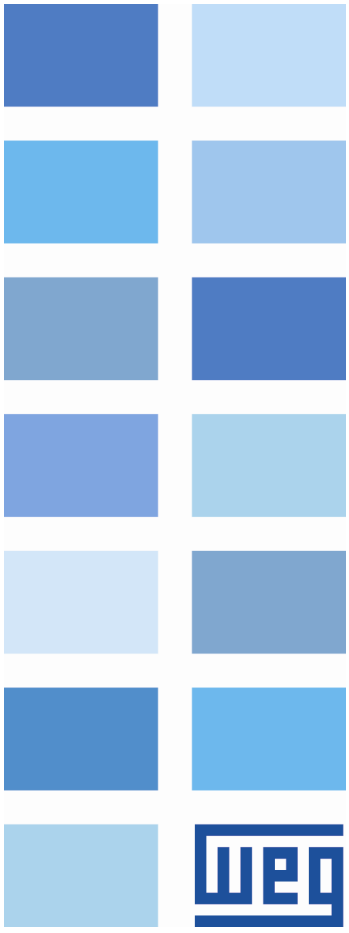
Software

WPS V1.20

Manual do Usuário



Manual do Usuário

Série: WPS

Idioma: Português

Nº do Documento: 10001027753 / R00

Data de Publicação: 11/2010

Sumário

1 WPS V1.20 8

1 Bem Vindo 8
2 Identificação 8
3 Visão Geral 8
4 Novidades 8
5 Contato 9
6 Suporte Técnico 9
7 Download 9
8 Informações Legais 9
9 Aviso 10

2 Instalação 11

1 Antes de Instalar 11
2 Requisitos Mínimos 11
3 Equipamentos Suportados 12
4 Java Runtime Environment 12
5 Instalando 13
6 Desinstalando 14
7 Gerenciador de Comunicação 14

3 Iniciando 15

1 Ambiente 15
2 Guia de Inicialização Rápida 15

4 Gerenciador de Comunicação 23

1 Executando 23
2 Configuração 23
3 Adicionar/Remover Conexões 23
4 Cabos 24
5 Conversor USB/Serial 25
6 Instalando Driver USB 26
7 Desinstalando Driver USB 27
8 Atualizando Driver USB 27
9 Verificando a Instalação do Driver USB 28

5 Editor Ladder 29

1 Programação Ladder 29
2 Editor de Variáveis/Ladder 31
3 Editor/Lista de Variáveis 31
4 Dados das Variáveis 33
5 Editor Ladder/Rungs 34
6 Editando uma Rung 35
7 Editando Título e Comentário da Rung 36
8 Inserindo Elementos na Rung 37
9 Navegação pelo Teclado na Rung 38
10 Navegação pelo Mouse na Rung 40
11 Inserindo Elementos em Série 42
12 Inserindo Elementos em Paralelo 43
13 Edição de Variáveis na Rung 43

6 Tarefas 46

1 Configuração de Tarefas 46

7 Editor de Telas 53

1 Configuração das Telas 53
2 Teclas de Função 55
3 Gerenciamento de Telas 55
4 Manipulação de Bits 56
5 Senhas 58
6 Alarmes 60
7 Alarmes Configuráveis 62
8 Alarmes Internos 63
9 Programação das Telas 64
10 Componente Text 68
11 Componente Numeric Input 69
12 Componente Numeric Output 70
13 Componente Bargraph 71
14 Componente Message 72

8 Monitoração 73

1 Monitoração Online Ladder 73

9 PLC300 75

1 Descrição 75
2 Tipo de Dados 75
3 Glossário 76

10 Linguagem Ladder 78

1 PLC300 78

V1.2X 78

Contact 78
NOCONTACT 78
NCCONTACT 78
PTSCONTACT 79
NTSCONTACT 80
Coil 81
DIRECTCOIL 81
NEGATEDCOIL 81
SETCOIL 82
RESETCOIL 83
TOGGLECOIL 83
IMMEDIATECOIL 84
Logic Bit 85
RESETBIT 85
SETBIT 86
TESTBIT 87
Logic Boolean 88
NOT 88
AND 88
OR 89
XOR 90
XNOR 91
Logic Shift 92
ASHL 92
ASHR 93
SHL 94
SHR 95
Logic Rotate 96
ROL 96
ROR 97
Math Basic 97
ABS 97
NEG 98
ADD 99
SUB 100
MUL 101
DIV 102
MOD 103
Math Extended 104
LN 104
EXP 104
LOG10 105
ALOG10 106
POW 107
SQRT 108
Math Trigonometry 108
SIN 108
COS 109
TAN 110
ASIN 110

ACOS 111
ATAN 112
ATAN2 113
Math Util 113
MAX 113
MIN 114
SAT 115
Compare 117
COMPEQ 117
COMPGE 118
COMPGT 119
COMPLE 120
COMPLT 121
COMPNE 122
Conversion To Bool 123
BYTE_TO_BOOL 123
WORD_TO_BOOL 123
DWORD_TO_BOOL 124
REAL_TO_BOOL 125
Conversion To Byte 126
BOOL_TO_BYTE 126
WORD_TO_BYTE 126
DWORD_TO_BYTE 127
REAL_TO_BYTE 128
Conversion To Word 129
BOOL_TO_WORD 129
BYTE_TO_WORD 129
DWORD_TO_WORD 130
REAL_TO_WORD 131
Conversion To Dword 132
BOOL_TO_DWORD 132
BYTE_TO_DWORD 132
WORD_TO_DWORD 133
REAL_TO_DWORD 134
Conversion To Real 134
BOOL_TO_REAL 134
BYTE_TO_REAL 135
WORD_TO_REAL 136
DWORD_TO_REAL 136
Conversion To BCD 137
BCD_TO_WORD 137
WORD_TO_BCD 138
Conversion Rad-Deg 138
RAD_TO_DEG 138
DEG_TO_RAD 139
Data Transfer 140
STORE 140
ILOADBOOL 141
ILOAD 142
ISTOREBOOL 143
ISTORE 144
MUX 145
DEMUX 147
SEL 149

SWAP 150
Counter 151
CTD 151
CTU 152
CTUD 153
Timer 155
TON 155
TOF 156
TP 157
Control 159
PID 159
Filter 162
LOWPASS 162
Hardware 163
READENC 163
READENC2 164
READENC3 165
PWM 166
IMMEDIATE_INPUT 167
IMMEDIATE_OUTPUT 168
RTC 169
INTIME 169
INWEEKDAY 171
Screen 172
SETSCREEN 172
CANopen 173
CANopen 173
CO_SDORead 175
CO_SDOWrite 177
CO_MasterControlStatus 179
CO_SlaveStatus 181
Modbus 182
Modbus 182
MB_ReadBinary 183
MB_ReadRegister 185
MB_WriteBinary 187
MB_WriteRegister 189
MB_MasterControlStatus 191
MB_SlaveStatus 193

Índice **195**

1 WPS V1.20

1.1 Bem Vindo

Bem vindo ao Software!

Esta documentação fornece uma visão geral do WPS V1.20 e contém toda a documentação disponível para o WPS V1.20.

Use o índice à esquerda para navegar pelos tópicos.

1.2 Identificação

WPS V1.20 - Manual do Usuário
Data de Publicação: 11/2010

1.3 Visão Geral

O Software ou WPS V1.20 é a mais nova geração de software da WEG, concebida para ser uma ferramenta integrada, agregando os conceitos de:

- Multi-Produtos, visando atender ampla gama de produtos da WEG; inicialmente estará atendendo o equipamento PLC300,
- Multi-Utilização visando permitir:
  - Configuração dos equipamentos,
  - Programação dos equipamentos em Linguagem Ladder,
  - Monitoração dos equipamentos, e
  - Assistente de criação e configuração de aplicações (futuro).

O ambiente de programação em Linguagem Ladder atende aos requisitos da NORMA IEC61131-3.

É uma ferramenta que proporciona flexibilidade ao usuário, pois lhe permite a programação de um ou mais recursos, dentro da mesma configuração.

1.4 Novidades

WPS V1.20

- Comunicação com equipamento PLC300 V1.2X,
- Programação de tarefas,
- Alteração no endereços de marcadores para compatibilidade com o Modbus,
- Ferramenta para importar/exportar projetos,

- Verificação da versão de firmware do PLC300 no download,
- Parar/executar programa como uma ação global,
- Persistência do ambiente ao sair,
- Melhorias na comunicação,
- Opções de download:
  - Inicializar variáveis retentivas,
  - Limpar histórico de alarmes,
  - Download do código fonte,
- Compatível com Windows 7 x64, Windows 7 x86, e
- Compatível com Windows Vista x64, Windows Vista x86.

1.5 Contato

Website: http://www.weg.net/

1.6 Suporte Técnico

Contate uma filial ou representante.

1.7 Download

Faça o download do WPS V1.20 da página de downloads em http://www.weg.net/.

Selecione o(s) arquivo(s) para fazer o download e os armazene em uma pasta no seu disco rígido.

Descompacte os arquivos.

Na pasta onde você descompactou os arquivos você encontrará o instalador. Dê um duplo clique neste icone para iniciar a instalação.

Para mais detalhes veja item Instalação > Instalando.

1.8 Informações Legais

Nota! Este programa de computador está protegido pelas leis dos direitos autorais e tratados internacionais. Sua reprodução ou distribuição, parcial ou total, sem autorização prévia pode resultar em severas penalidades civis e criminais, sujeito às sanções previstas em lei.

1.9 Aviso

Perigo! Uso deste software pode alterar a operação ou desempenho do drive. O usuário é responsável pela adoção de todas as precauções necessárias para garantir a segurança do equipamento e do pessoal envolvido. Antes de aplicar este Software, leia atentamente as Instruções da Ajuda Online. A não observação destas instruções pode causar sérios danos no equipamento e resultar em graves lesões corporais.

2 Instalação

2.1 Antes de Instalar

Verifique os seguintes itens antes de instalar o WPS V1.20:

- Se o microcomputador possui os Requisitos Mínimos,
- Se a versão do WPS V1.20 é compatível com seu equipamento, veja Equipamentos Suportados, e
- O Java Runtime Environment V1.6.0 ou superior precisa estar instalado no seu computador. Instale o JRE antes de instalar o WPS V1.20.

2.2 Requisitos Mínimos

Item	Descrição
Processador	PC com processador mínimo Pentium IV Intel(R) Core(TM)2 DUO ou superior é recomendado
Plataforma	Windows XP x86 SP3
	Windows XP x64 SP1
	Windows Vista x86 SP1
	Windows Vista x64 SP1
	Windows 7 x86
	Windows 7 x64
	Você deve ter privilégios de administrador para instalar o software
Memória	512 MB para Windows XP, 1 GB ou mais é recomendado
	1 GB para Windows Vista, 2 GB ou mais é recomendado
	1 GB para Windows 7, 2 GB ou mais é recomendado
Cor do Monitor	High color (16 bits) ou melhor
Resolução do Monitor	1024x768 pixels ou resolução superior é recomendado
Espaço em Disco	100 MB de espaço disponível no disco para WPS
	150 MB de espaço disponível no disco para Java Runtime Environment (depende da versão)
Unidade de Disco	Unidade CD-ROM ou DVD-ROM
Comunicação	Interface serial RS232 (ponto-a-ponto)
	Comunicação USB (ponto-a-ponto)
Dispositivo Apontador	Mouse ou dispositivo apontador compatível
Java Runtime Environment	Java Runtime Environment (JRE) 6.0 ou superior
Navegador	Mozilla Firefox 3.6 ou superior
	Opera 10 ou superior
	Internet Explorer 7 ou superior
	JavaScript habilitado Navegador deve suportar Cascading Style Sheets (CSS1) e JavaScript

2.3 Equipamentos Suportados

A seguinte tabela apresenta os equipamentos e versões de firmware que cada base de dados fornecida com o WPS suporta.

Versão do WPS	PLC300
WPS V1.20	V1.20
WPS V1.01	V1.00 * (Versão obsoleta)
WPS V1.00	V1.00 * (Versão obsoleta)

* Recomendamos fazer upgrade do firmware do PLC300 para V1.20.

2.4 Java Runtime Environment

O Java Runtime Environment (JRE) V1.6.0 ou superior precisa ser instalado no seu computador.

Para obter a última versão do JRE gratuitamente, vá para http://www.java.com/.

Para verificar se você tem o JRE instalado ou qual versão que você possa já ter, veja abaixo.

Verifique a versão do JRE no seu Computador Usando o Prompt de Comando

Para verificar se você tem o JRE correto digite o seguinte comando no prompt de comando ou num terminal.

java -version

Se você receber uma mensagem de erro indicando 'java' como um comando desconhecido então você precisa fazer o download do JRE.

Se você receber uma resposta ao comando que indica que o número da versão do Java é menor que 1.6.0, tal como 1.2, 1.3, 1.4.1, 1.4.2, 1.5.0, etc, então você precisa fazer o download de uma versão mais nova do JRE.

Verifique a versão do JRE no seu Computador Usando o Sítio Web Java.com

Opção 1:

Existe uma outra opção para verificar a instalação do software Java.

No endereço http://www.java.com/ há um link chamado Tenho o Java? onde você pode encontrar instruções para verificar o JRE instalado no seu computador.

Quando este manual foi escrito o link era http://www.java.com/pt_BR/download/installed.jsp.

Siga as instruções nesta página.

Opção 2:

Você também pode utilizar o seguinte link (página em inglês) para verificar a versão do java: http://www.java.com/en/download/help/testvm.xml .

Instruções de Instalação

Siga as intruções em http://www.java.com/.

Mais Informações

Mais detalhes sobre Java Runtime Environment podem ser encontrados em:

- http://www.java.com/

2.5 Instalando

Ao instalar uma versão mais nova do WPS é recomendado primeiro remover a versão anterior e reinicializar o sistema para ter certeza que a nova instalação será executada corretamente.

Fechar todos os programas abertos atualmente em execução para evitar interferência com o processo de instalação.

Para instalar o WPS V1.20, por favor seguir os passos abaixo.

Instalação

- Dar um duplo clique no ícone do instalador para iniciar o instalador. O programa de instalação iniciará e uma janela de boas vindas aparecerá.
- No assistente de instalação, responder à Licença de Uso, então pressionar botão Avançar.
- Especificar uma pasta vazia dentro da qual será instalada o WPS V1.20. Certificar-se que a localização da instalação está correta e que existe espaço suficiente em disco para a instalação.
- Confirmar para criar uma pasta vazia.
- Selecionar o tipo de instalação que desejar.
- Selecionar o grupo de programa para criar os atalhos do WPS V1.20.
- Selecionar os atalhos de programas adicionais: criar um atalho no desktop e criar um atalho para inicialização rápida.
- Por favor revisar ou alterar qualquer uma das configurações. Clicar no botão

Instalar para iniciar a instalação.
- Clicar no botão Concluir para concluir o setup.

2.6 Desinstalando

Se necessário, remover o WPS V1.20 usando os procedimentos abaixo.

Desinstalação

> Nota! Conforme indicado abaixo, usar o utilitário Adicionar ou Remover Programas para remover o WPS V1.20. Não apagar arquivos e pastas manualmente.

- Do menu Iniciar, selecionar Painel de Controle.
- Dar um duplo clique no ícone Adicionar ou Remover Programas.
- Selecionar WPS V1.20 da lista e clicar no botão Remover.
- Seguir as instruções para remover o software.

2.7 Gerenciador de Comunicação

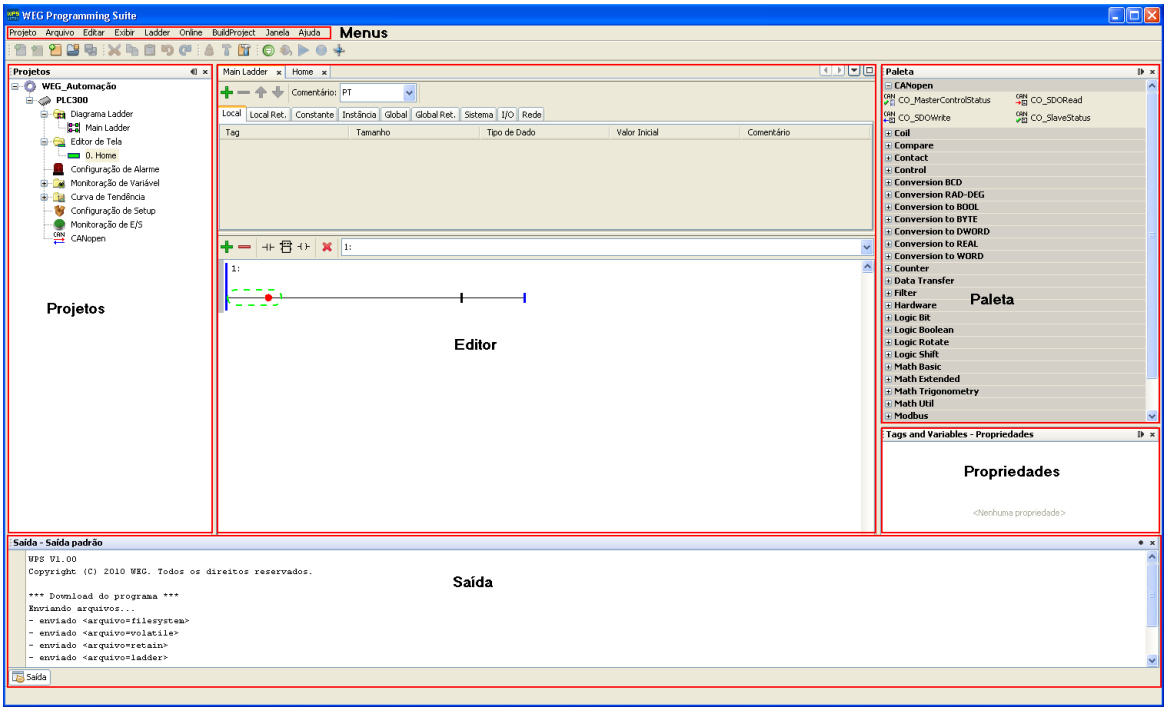
O Gerenciador de Comunicação é instalado automaticamente quando o WPS V1.20 é instalado.

3 Iniciando

3.1 Ambiente

O Software é dividido em 6 campos principais:

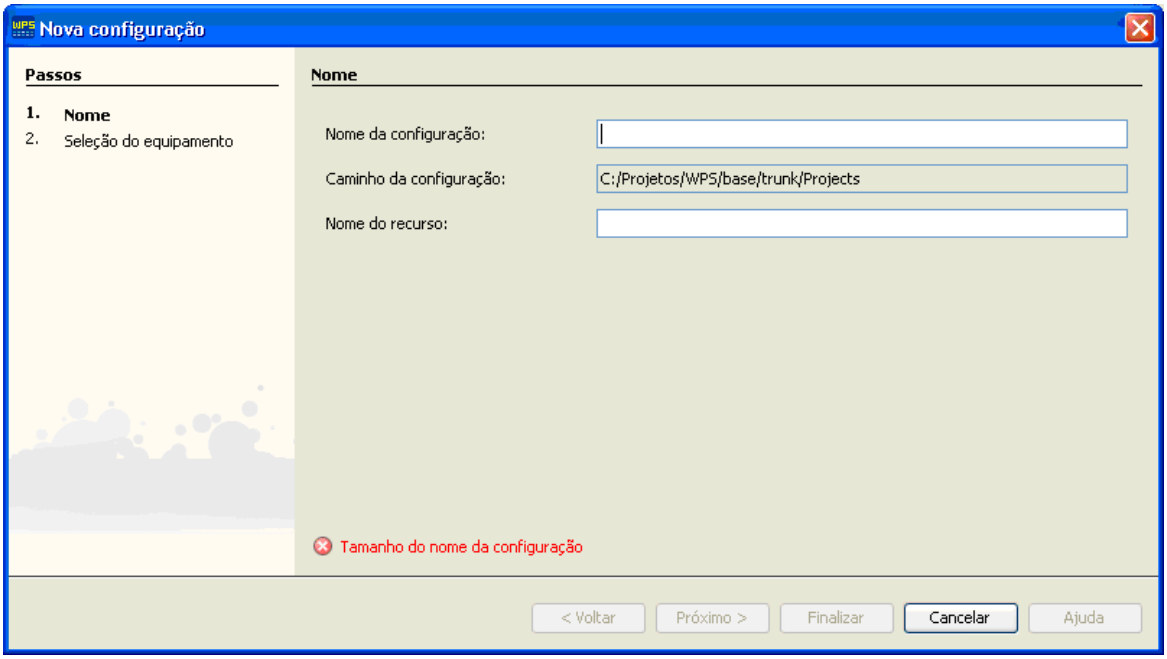
- Menus - Exibe as diversas opções de edição, visualização, comunicação e ajuda do ambiente de desenvolvimento,
- Janela de Projetos - Exibe a árvore contendo a estrutura da Configuração aberta,
- Janela do Editor - Parte principal do ambiente de desenvolvimento, onde ocorre a edição dos componentes que irão formar o arquivo fonte,
- Janela da Paleta - Contem os componentes a serem inseridos no editor através do drag'n drop,
- Janela de Propriedades - Exibe as propriedades do componente selecionado, e
- Janela de Saída - Exibe mensagens de compilação e download.



3.2 Guia de Inicialização Rápida

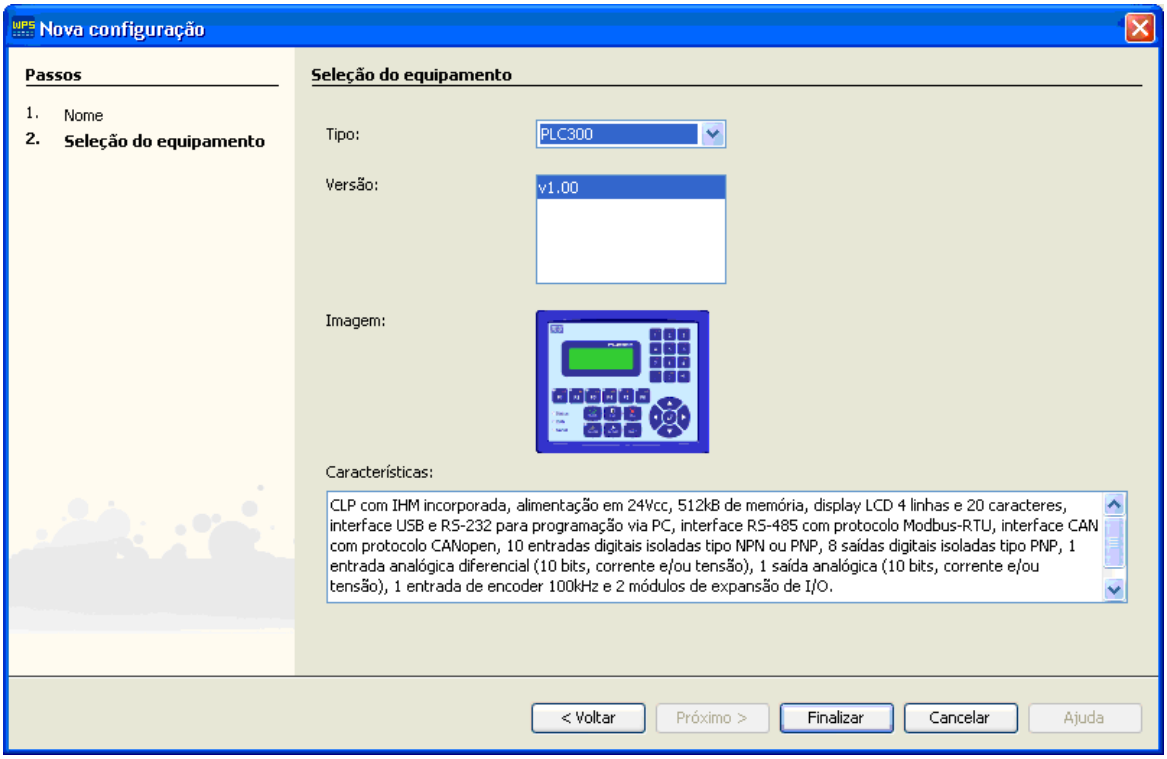
Criação da Configuração

Para criar uma configuração é preciso selecionar a opção "Projeto > Nova Configuração" no menu ou pressionar as teclas de atalho "Ctrl+Shift+C". Uma janela para configurar o nome da Configuração (Projeto ou Planta) e do Recurso (Equipamento) irá abrir.

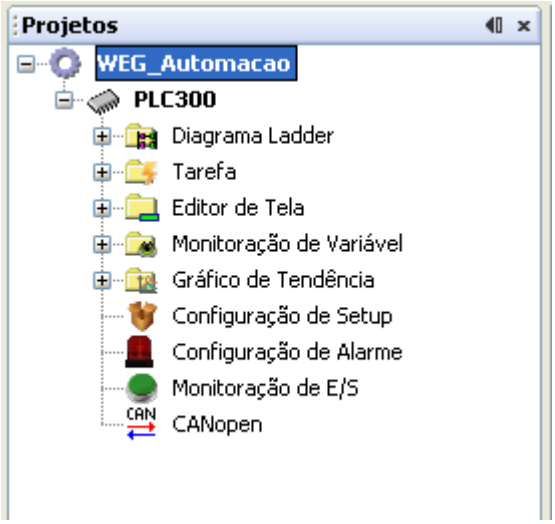


Insira nomes válidos e selecione a botão próximo.

Uma janela contendo os equipamentos suportados pelo ambiente de programação irá aparecer. Selecione o equipamento e a versão desejados e selecione o botão finalizar.


Nova configuração
Passos
1. Nome
2. Seleção do equipamento
Seleção do equipamento
Tipo:
PLC300
Versão:
v1.00
Imagem:
Características:
CLP com IHM incorporada, alimentação em 24Vcc, 512kB de memória, display LCD 4 linhas e 20 caracteres, interface USB e RS-232 para programação via PC, interface RS-485 com protocolo Modbus-RTU, interface CAN com protocolo CANopen, 10 entradas digitais isoladas tipo NPN ou PNP, 8 saídas digitais isoladas tipo PNP, 1 entrada analógica diferencial (10 bits, corrente e/ou tensão), 1 saída analógica (10 bits, corrente e/ou tensão), 1 entrada de encoder 100kHz e 2 módulos de expansão de I/O.
< Voltar
Próximo >
Finalizar
Cancelar
Ajuda

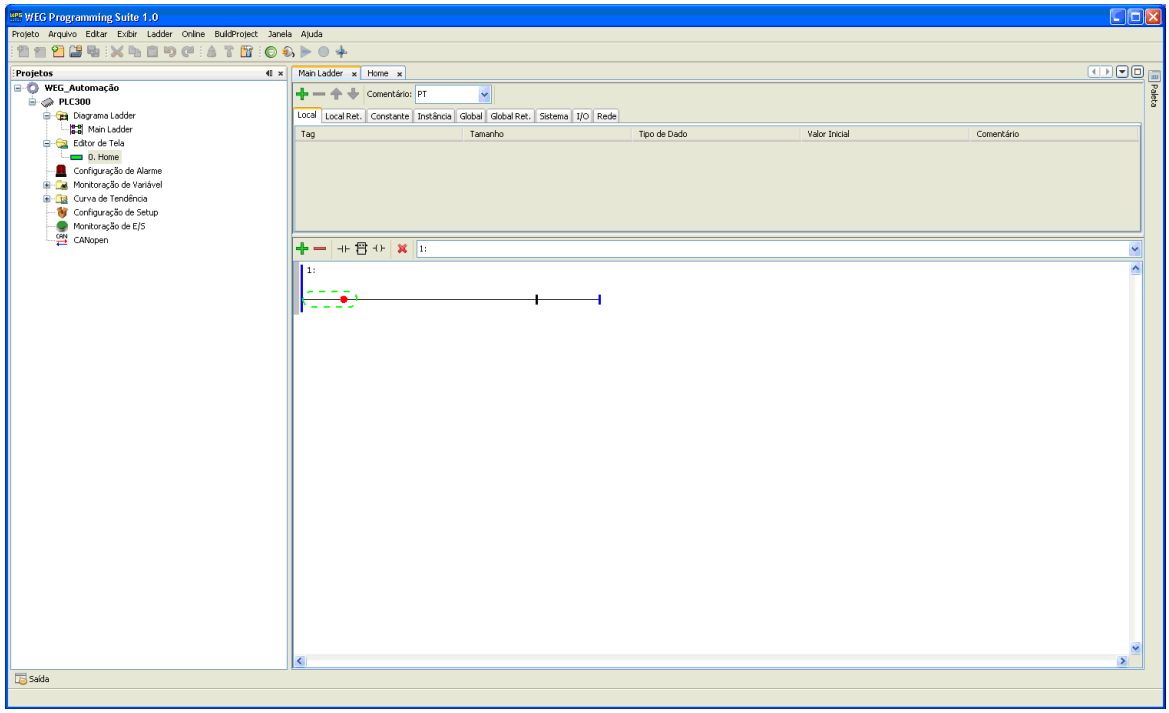
O projeto criado deverá conter a seguinte estrutura:



Edição do Programa

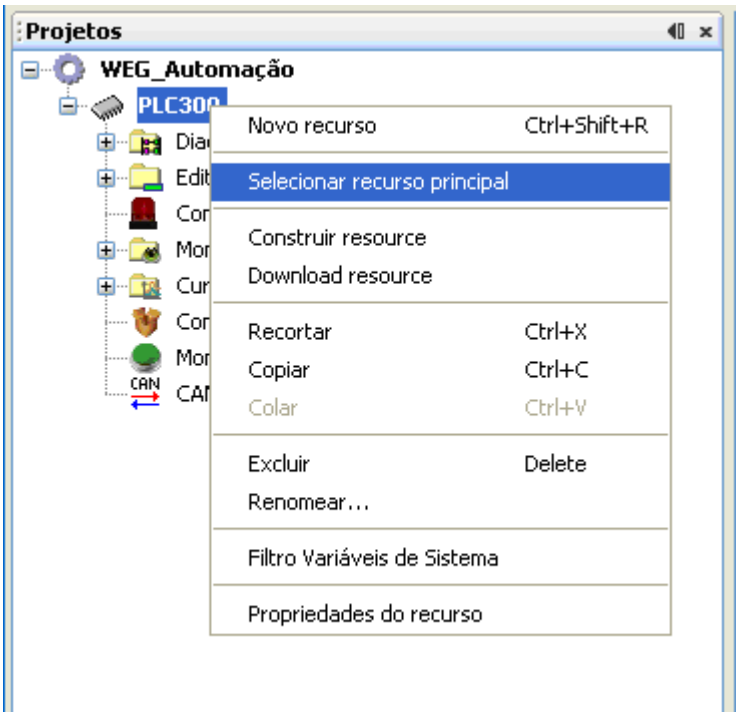
Após a criação da configuração e do recurso o usuário deve editar programa conforme a necessidade da aplicação.

Para maiores detalhes consulte Editor Ladder e Editor de Telas.

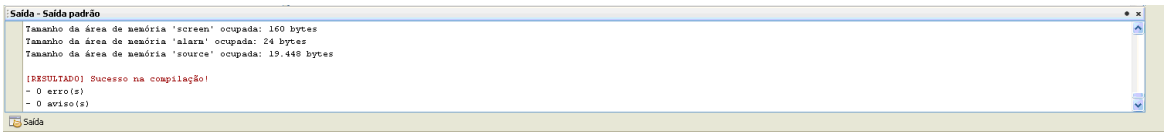


Compilar Recurso

Para compilar o recurso, o usuário deverá selecionar a opção "Construir>Construir recurso principal" no menu ou pressionar a tecla de atalho "F4". Para isso, certifique-se de que o recurso a ser compilado é o recurso selecionado na janela de projetos. O recurso selecionado tem sua formatação em negrito. Caso o recurso principal não seja o recurso desejado deve-se clicar com o botão direito no recurso e selecionar a opção "Selecionar recurso principal".



O recurso será compilado e no final deverá aparecer a mensagem: "[RESULTADO] Sucesso na compilação!" na janela de Saída, caso contrário verifique as mensagens de erro geradas e faça as correções necessárias no editor.


Saída - Saída padrão
Tamanho da área de memória 'screen' ocupada: 160 bytes
Tamanho da área de memória 'alarm' ocupada: 24 bytes
Tamanho da área de memória 'source' ocupada: 19.448 bytes
[RESULTADO] Sucesso na compilação!
- 0 erro(s)
- 0 aviso(s)
Saída

Execução do Gerenciador de Comunicação (Local)

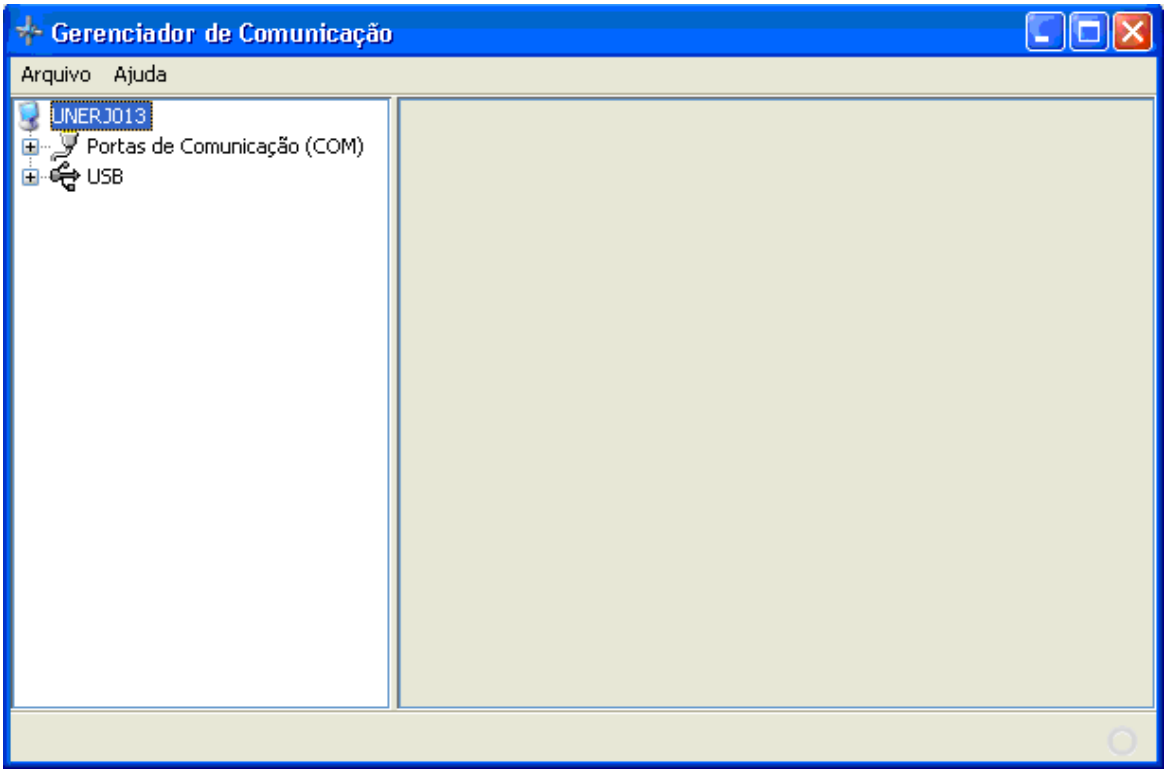
Antes de iniciar o download é necessário que o gerenciador de comunicação seja ativado.

Para ativar o gerenciador de comunicação o usuário deve selecionar a opção

"Online>Gerenciador de comunicação local".

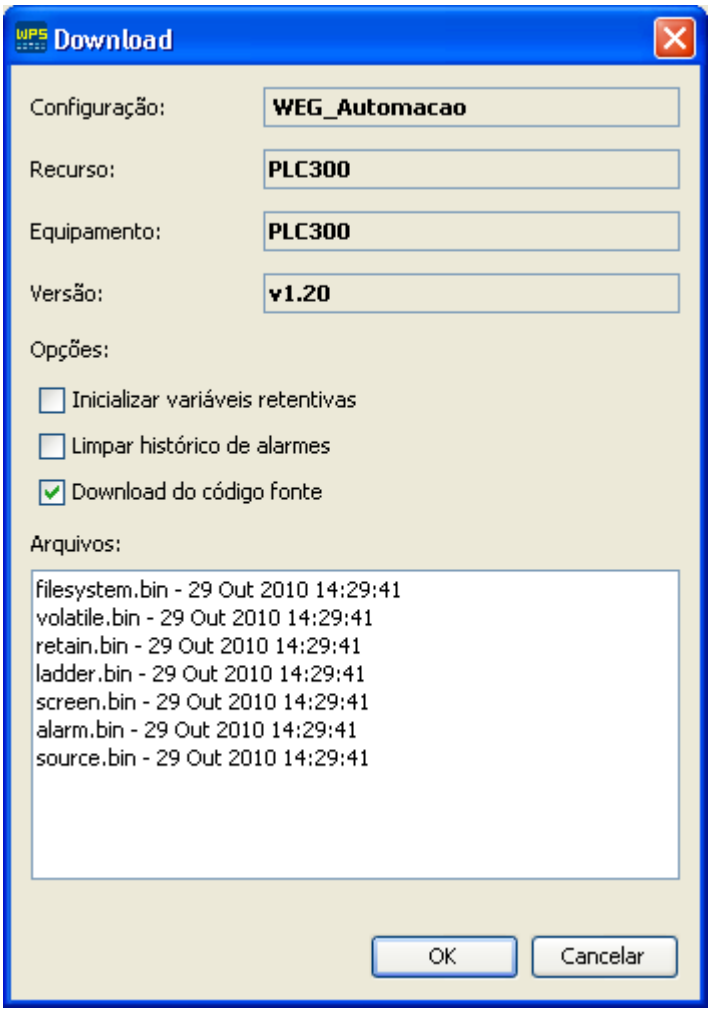
Uma tela com as configurações do gerenciador irá abrir. Configure a conexão com o equipamento, maiores detalhes ver Gerenciador de Comunicação.

FIGURA7

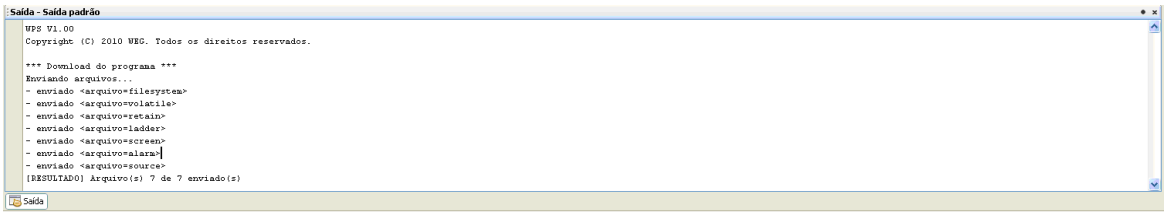


Download do Recurso

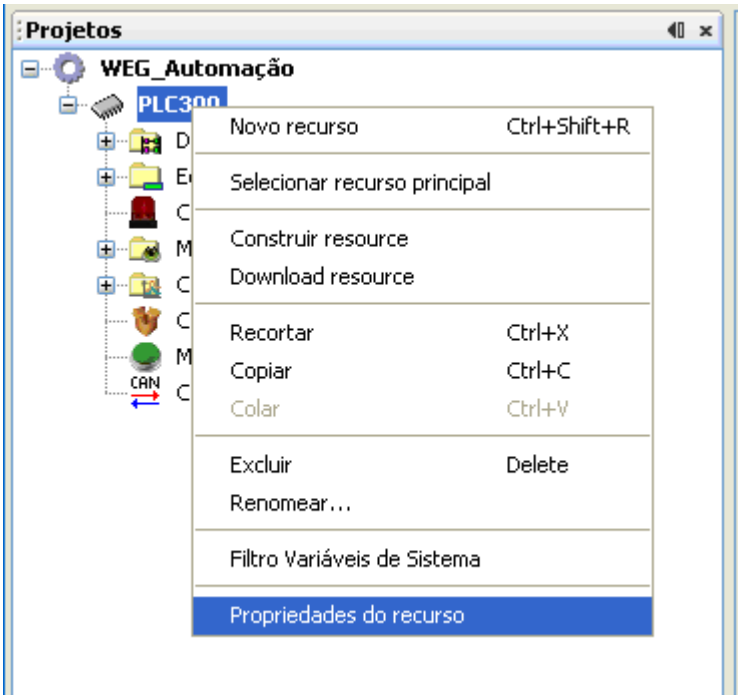
Para realizar o download do recurso o usuário deve selecionar a opção "Online>Download do recurso principal". Uma janela contendo as informações do download deverá aparecer.



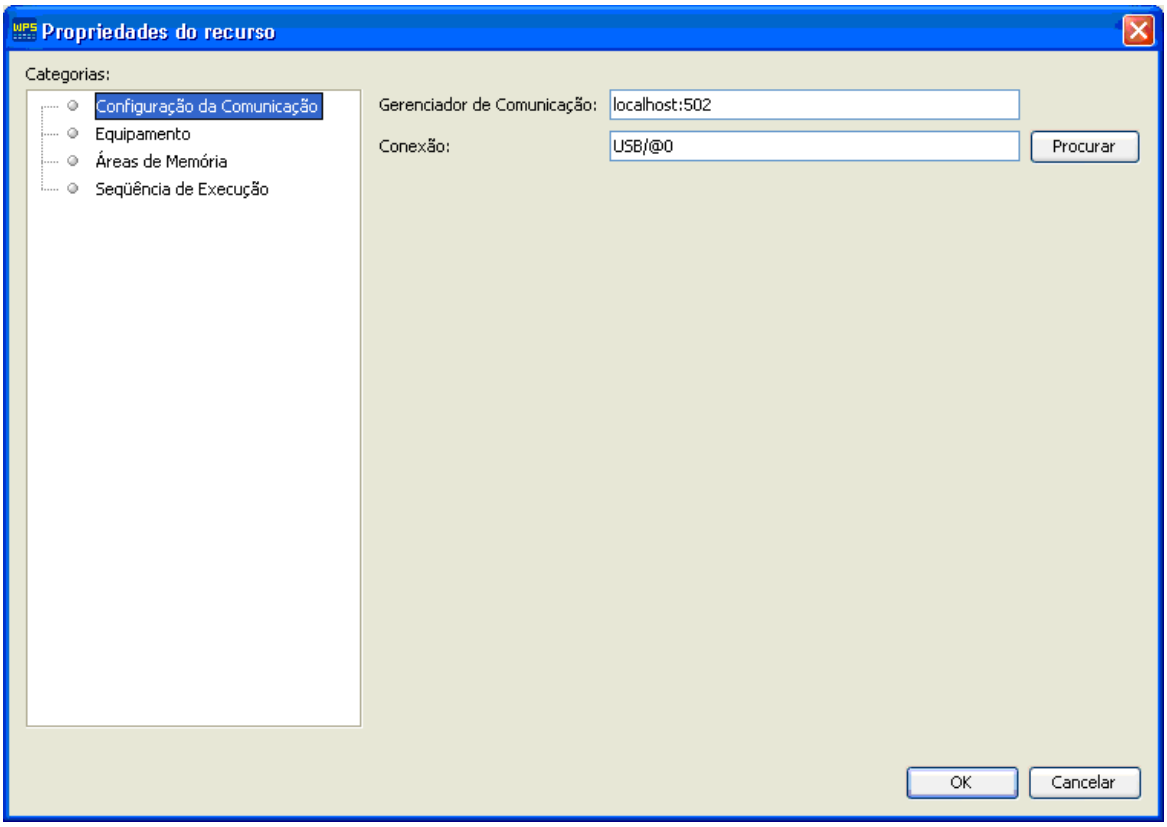
Verifique se as informações correspondem ao equipamento conectado. A ação de download irá parar a execução do programa do usuário e reiniciá-lo logo após a conclusão do download. No final do download deverá aparecer uma mensagem "[RESULTADO] Arquivo(s) 7 de 7 enviado(s)". Caso tenha acontecido alguma falha durante o envio, verifique as propriedades de comunicação.



No Software estas propriedades podem ser visualizadas clicando com o botão direito sobre o recurso e selecionando a opção "Propriedades do recurso".



Uma janela contendo as propriedades do recurso irá aparecer. Selecione a opção "Configuração da comunicação". Verifique se as opções estão configuradas corretamente.


Propriedades do recurso
Categorias:
Configuração da Comunicação
Equipamento
Áreas de Memória
Seqüência de Execução
Gerenciador de Comunicação:
localhost:502
Conexão:
USB/@0
Procurar
OK
Cancelar

4 Gerenciador de Comunicação

4.1 Executando

Para executar o gerenciador de comunicação na máquina local basta clicar no ícone disponível no Toolbar.

O comando para executar o gerenciador de comunicação também está disponível no menu "Online" opção "Gerenciador de comunicação local".

4.2 Configuração

Configurações para o Gerenciador de Comunicação

- Ativo: ativa/desativa a comunicação do Gerenciador de Comunicação com as aplicações que se comunicam com os equipamentos através dele,
- Número da Porta: número da porta de comunicação TCP/IP que o Gerenciador de Comunicação utiliza para comunicação com as aplicações que se comunicam com os equipamentos através dele,
- Autenticação de IP: lista com os endereços IPs que podem acessar o Gerenciador de Comunicação. Uma lista vazia indica que qualquer endereço IP pode acessar o Gerenciador de Comunicação.

4.3 Adicionar/Remover Conexões

Para adicionar uma nova conexão, selecione na árvore a interface a qual o equipamento está conectado.

Interface Portas de Comunicação

- Expanda o item da porta COM desejada.
- Altere as configurações da porta COM conforme as utilizadas pelo equipamento.
- Selecione o protocolo de comunicação (Modbus-RTU).
- Altere as configurações do protocolo conforme o desejado.
  - Atraso de transmissão (ms): aguarda este tempo antes de enviar o próximo telegrama.
    Faixa de valores: 0 ... 20000
    Padrão: 0
  - Atraso na resposta (ms): aguarda este tempo antes de tentar receber a resposta ao telegrama enviado.
    Faixa de valores: 0 ... 20000
    Padrão: 0
  - Timeout (ms): aguarda este tempo antes de indicar que a resposta não foi recebida.
    Faixa de valores: 100 ... 20000

Padrão: 5000
- Clique em nova conexão...
  - Selecione o endereço serial para a nova conexão conforme o configurado no equipamento.

Interface USB

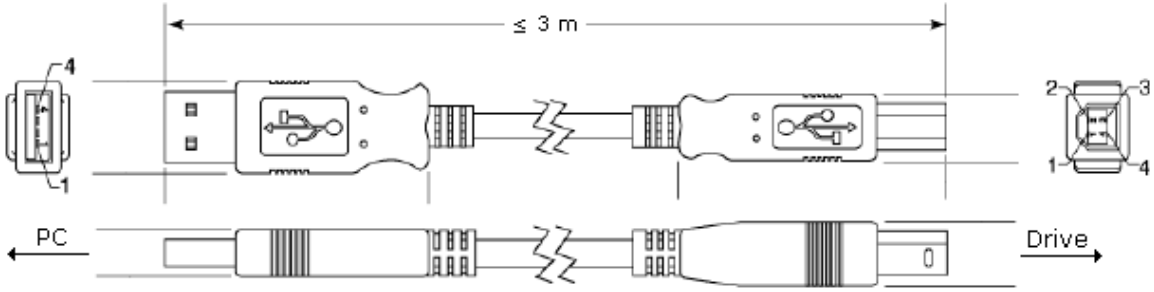
- Selecione o item USB.
  - Altere as configurações conforme o desejado.
  - Atraso de transmissão (ms): aguarda este tempo antes de enviar o próximo telegrama.
    Faixa de valores: 0 ... 20000
    Padrão: 0
  - Atraso na resposta (ms): aguarda este tempo antes de tentar receber a resposta ao telegrama enviado.
    Faixa de valores: 0 ... 20000
    Padrão: 0
  - Timeout (ms): aguarda este tempo antes de indicar que a resposta não foi recebida.
    Faixa de valores: 100 ... 20000
    Padrão: 5000
- Clique em nova conexão...

4.4 Cabos

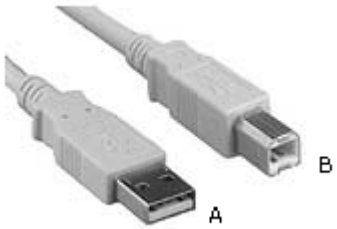
Consulte o manual do usuário do PLC300.

Cabo USB

A figura abaixo apresenta o detalhamento do cabo para a conexão USB ponto-a-ponto.



A figura abaixo apresenta os conectores.



A = Conector Tipo A
B = Conector Tipo B

Tamanho máximo do cabo: 3 metros.

Nota! Utilize sempre cabo de interconexão USB blindado, "standard host/ device shielded USB cable". Cabos sem blindagem podem provocar erros de comunicação.

Nota! A conexão USB é isolada galvânicamente da rede elétrica de alimentação e de outras tensões elevadas internas ao drive. A conexão USB, porém, não é isolada do terra de proteção (PE). Usar laptop isolado para ligação ao conector USB ou desktop com conexão ao mesmo terra de proteção (PE) do drive.

Sugestões de Compra

Fabricante:
- Samtec, Inc: http://www.samtec.com/

Se você desejar comprar um cabo USB diretamente da Samtec, por favor veja abaixo.

Descrição	Item
Cabo USB blindado de alta velocidade Revisão 2.0, 1 m, Samtec	USBC-AM-MB-B-B-S-1
Cabo USB blindado de alta velocidade Revisão 2.0, 2 m, Samtec	USBC-AM-MB-B-B-S-2
Cabo USB blindado de alta velocidade Revisão 2.0, 3 m, Samtec	USBC-AM-MB-B-B-S-3

No momento em que este manual foi escrito, a especificação era encontrada em http://www.samtec.com/ftppub/cpdf/USBC-AM-BM-B-B-S-X-MKT.pdf

4.5 Conversor USB/Serial

O conversor USB/SERIAL é a melhor solução para quem deseja conectar equipamentos seriais (RS232) em portas USB, isto é, para conectar a saída USB do microcomputador a um equipamento serial RS-232.

É uma solução de baixo custo que resolve a necessidade de instalar novas portas seriais em microcomputadores que possuam todos os barramentos ocupados ou em equipamentos (notebooks) que não possuem portas RS232.

O conversor USB/SERIAL permite a conexão plug & play com seu microcomputador, deixando livre a porta serial existente.

A figura abaixo apresenta exemplos deste conversor.

Sugestões de Compra no Brasil

Fabricante: Leadership Comércio e Importação Ltda - código do produto 8430.
Fabricante: Clone - código do produto 05087.
Fabricante: GoldShip Computer Accessories - código do produto 0470.

Para maiores informações sobre a instalação, consulte o manual do conversor USB/ SERIAL que acompanha o produto.

4.6 Instalando Driver USB

Instalação

Utilize o instalador disponível do driver USB.

Feche todas as aplicações no seu PC. Se você está usando um software anti-virus ou firewall, feche-os (ou desabilite suas funções).

Inicie o instalador e siga os menus.

O instalador realiza uma pré-instalação do software do driver. Durante este processo, o driver se torna conhecido do sistema. No Windows XP o arquivo INF é copiado para a pasta INF e um arquivo INF pré-compilado é criado.

No Windows Vista e no Windows 7 o pacote do driver é armazenado no local de armazenamento de drivers. Quando um dispositivo PnP é conectado ao sistema, ele encontra o driver relacionado e instalá-lo.

No final da instalação é solicitado ao usuário para desconectar e conectar o equipamento.

Durante a instalação do driver uma entrada no painel de controle de programas do windows é criado. Esta entrada pode ser usado para desinstalar o software do driver.

Verificação da Instalação

Você pode verificar se a instalação foi bem sucedida no tópico Verificando a Instalação do Driver USB.

4.7 Desinstalando Driver USB

Se você precisa desinstalar o driver USB do PC, siga o procedimento descrito abaixo.

Desinstalação

O desinstalador remove a instalação do driver realizado anteriormente. Durante o processo de remoção o driver é desinstalado e as entradas do registro correspondente e pré-instalação do software de driver são removidos. Após esta etapa o sistema se comporta de tal maneira como se o software do driver nunca tivesse sido instalado.

4.8 Atualizando Driver USB

Se você precisa atualizar o driver USB do PC, siga o procedimento descrito abaixo.

Atualização

A atualização de driver é uma combinação de a remoção do software de driver e uma instalação pela primeira vez. O instalador detecta automaticamente se uma atualização ou uma instalação pela primeira vez é necessário.

Utilize sempre o instalador de driver disponível.

Verificação da Instalação

Você pode verificar se a instalação foi bem sucedida no tópico Verificando a Instalação do Driver USB.

4.9 Verificando a Instalação do Driver USB

Você pode verificar se a instalação foi bem sucedida no gerenciador de dispositivos (o equipamento precisa estar conectado ao PC).

Para executar o gerenciador de dispositivos, clique Iniciar , clique Executar, digite devmgmt.msc, e então clique OK. O gerenciador de dispositivos também pode ser acessado por Configurações > Painel de Controle > Sistema > Hardware > Gerenciador de Dispositivos,

No gerenciador de dispositivos, próximo ao final da lista você deve encontrar a entrada WEG USB devices contendo WEG USBIO V6.00. Isto indica que a instalação foi realizada com sucesso.

Verifique com o fabricante se existe algum driver USB mais novo.

5 Editor Ladder

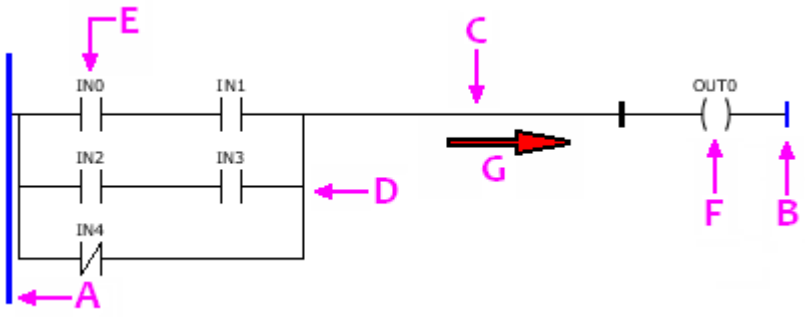
5.1 Programação Ladder

Programação Ladder é a representação gráfica de equações booleanas, combinando contatos (argumentos de entradas) com bobinas (resultados de saída).

O programa em Ladder possibilita testar e modificar dados por símbolos gráficos padrões. Estes símbolos são posicionados no diagrama ladder de maneira semelhante a uma linha de um diagrama lógico com relés. O diagrama Ladder é delimitados na esquerda e na direita por linhas de barramento.

Componentes Gráficos

Os componentes gráficos básicos de um diagrama Ladder são mostrados abaixo.



A - Barramento esquerdo
B - Barramento direito
C - Ligação horizontal
D - Ligação vertical
E - Contato
F - Bobina
G - Fluxo de potência

Barramentos

O editor é delimitado na esquerda por uma linha vertical conhecida como barramento esquerdo, e na direita por uma linha vertical conhecida como barramento direito.

Elementos de Ligação e Estados

Os elementos de ligação podem ser horizontal ou vertical. O estado dos elementos de ligação podem ser denotados por 1 ou 0, correspondendo ao valor Booleano literal 1 ou 0, respectivamente. O termo estado da ligação tem que ser sinônimo do termo fluxo da potência.

O estado do barramento esquerdo pode ser considerado sempre 1. Nenhum estado é definido no barramento direito.

Um elemento de ligação horizontal tem que ser indicado por uma linha horizontal. Um elemento de ligação horizontal transmite o estado do elemento imediatamente a esquerda para o elemento imediatamente a direita.

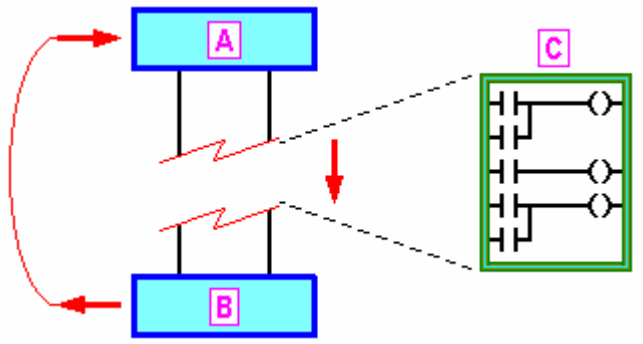
Um elemento de ligação vertical tem que consistir de linhas verticais intersectadas por uma ou mais ligações horizontais em cada lado. O estado da ligação vertical deverá representar o OU lógico dos estados 1 da ligações horizontais no lado esquerdo, isto é, o estado das ligações verticais deverá ser:

- 0 se o estado de todas as ligações horizontais incluídas na sua esquerda são 0,
- 1 se o estado de uma ou mais ligações horizontais incluídas na sua esquerda são 1.

O estado das ligações verticais tem que ser copiados para todas as ligações horizontais associados à sua direita. O estado das ligações verticais não pode ser copiado para as ligações horizontais associadas à sua esquerda.

Controle de Execução

A figura a seguir mostra como o programa em Ladder é executado. O cartão executa continuamente um ciclo de Varredura. O ciclo começa com o Sistema de E/S do hardware, compilando os últimos valores de todos os sinais de entrada e gravando seus valores em regiões fixas da memória.

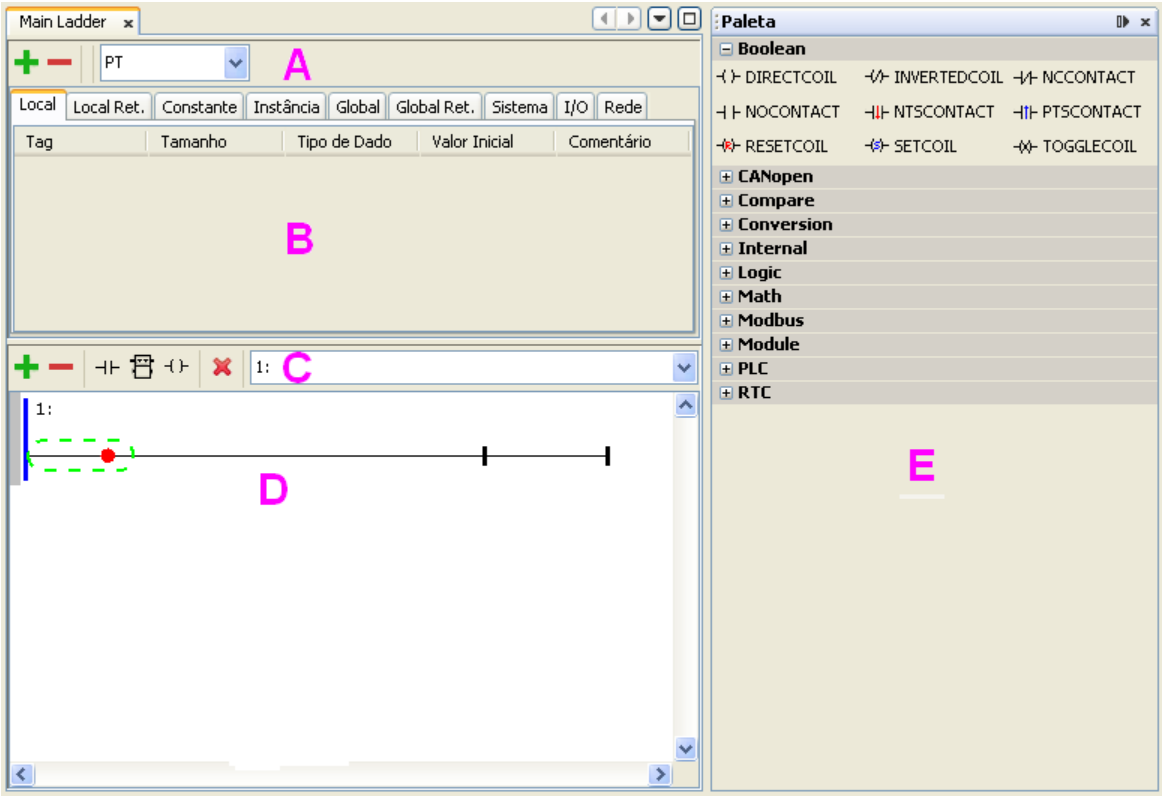


A - Entradas lidas para a memória
B - Memória escrita nas Saídas
C - Varredura das linhas do ladder

As linhas do programa ladder são então executados num ordem fixa, iniciando com a primeira linha. Durante a varredura do programa, novos valores das saídas físicas., como determinadas a partir da lógica das várias linhas do ladder, são inicialmente inscritos numa região da memória de saída. Finalmente, quando o programa ladder terminou a execução, todos os valores de saída retidos na memória são inscritos nas saídas físicas pelo hardware numa única operação.

5.2 Editor de Variáveis/Ladder

Sempre que uma arquivo ladder for aberto através do seu atalho na árvore de projetos o editor variáveis/ladder aparecerá conforme figura a seguir.



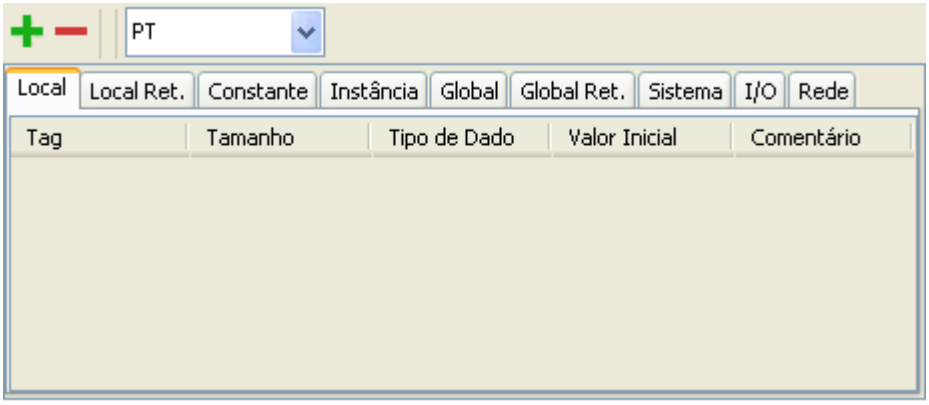
O editor variáveis/ladder possui os seguintes componentes:

A - Barra de ferramentas editor/lista de variáveis
B - Editor/lista de variáveis
C - Barra de ferramentas editor ladder/rungs
D - Editor ladder/rungs
E - Paleta de componentes ladder*

* Caso não esteja visível ativar pelo menu Janela opção Paleta

5.3 Editor/Lista de Variáveis

Durante a edição do ladder é necessário definir as variáveis utilizadas nos componentes ladder, essa definição pode ser feita diretamente no ladder e/ou no editor/lista de variáveis que estará sempre disponível na parte superior do ladder e terá a seguinte forma.



Através da barra de ferramentas do editor/lista de variáveis é possível executar as seguintes operações:

- Adicionar uma nova variável

 - Remover a variável selecionada

- Selecionar o idioma relativo aos comentários das variáveis

As variáveis são divididas em grupos de acordo com a aba selecionada na lista de variáveis conforme figura a seguir.

São possíveis os seguintes grupos:

Local: variáveis acessíveis somente pelo ladder que está sendo editado
Local Retentivo: semelhante ao grupo local mais com memória retentiva*
Constante: variáveis que armazenam valores constantes
Instância: variáveis utilizadas nas estruturas de dados dos blocos funcionais
Global: variáveis acessíveis por todos os arquivos ladder e por todas as telas de IHM
Global Retentivo: semelhante ao grupo global mais com memória retentiva*
Sistema: variáveis previamente definidas com funções do sistema
I/O: entrada e saídas físicas do equipamento
Rede: variáveis previamente definidas para comunicação em rede

* Uma memória retentiva mantém seu valor mesmo com o equipamento desligado

Para os User Function Block (Bloco de Função definido pelo usuário) existirão outros grupos conforme figura a seguir.

Grupos exclusivos dos User Function Block:

Input: variáveis de entrada do User Function Block
Output: variáveis de saída do User Function Block
Input Output: variáveis de entrada e saída do User Function Block

5.4 Dados das Variáveis

Ao definir uma variável através do editor/lista de variáveis alguns dados deverão ser definidos para as variáveis. Nos itens a seguir serão apresentados esses dados de acordo com o grupo que a variável pertence.

- Local e Local Retentivo:

Tag	Tamanho	Tipo de Dado	Valor Inicial	Comentário
VARIABLE0	0	BOOL	0	

Tag: identificação da variável
Tamanho: número de elementos do array* relacionado a variável
Tipo de Dado: tipo númerico da variável
Valor Inicial: valor que será carregado para variável durante a inicialização do equipamento
Comentário: comentário da variável no idioma selecionado

* Quando tamanho > 0 as variáveis são acessadas no ladder através do seu índice de array

- Constante:

Tag	Tipo de Dado	Valor	Comentário
VARIABLE1	BOOL	0	

Tag: identificação da variável
Tipo de Dado: tipo númerico da variável
Valor: valor constante da variável
Comentário: comentário da variável no idioma selecionado

- Global e Global Retentivo

Tag	Tamanho	Tipo de Dado	At	Endereço	Bit	Valor Inicial	Comentário
VARIABLE2	0	BOOL	%MB	0	0	0	

Tag: identificação da variável
Tamanho: número de elementos do array* relacionado a variável
Tipo de Dado: tipo númerico da variável
At: define qual área de memória global a variável acessa
Endereço: endereço relacionado a área de memória global
Bit: para tipo de dados booleanos e necessário definir o bit que a mesma acessa (0...7)
Valor Inicial: valor que será carregado para variável durante a inicialização do equipamento
Comentário: comentário da variável no idioma selecionado

* Quando tamanho > 0 as variáveis são acessadas no ladder através do seu índice de array

- Instância

Tag	Instância	Comentário
INST0	TON	

Tag: identificação da variável
Instância: tipo da estrutura de dados relacionada ao bloco funcional que a mesma será utilizada
Comentário: comentário da variável no idioma selecionado

- Sistema

Tag	Tipo de Dado	At	Endereço	Bit	Comentário
FIRMWARE	WORD	%SW	3000		Versão do firmware

Tag: identificação da variável
Tipo de Dado: tipo númerico da variável
At: define qual área de memória global a variável acessa
Endereço: endereço relacionado a área de memória global
Bit: para tipo de dados booleanos e necessário definir o bit que a mesma acessa (0...7)
Comentário: comentário da variável no idioma selecionado

- I/O

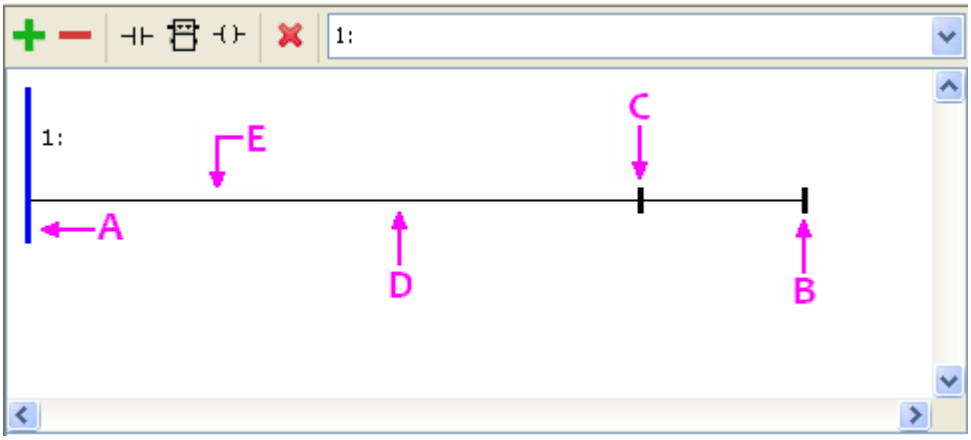
Tag	Tamanho	Tipo de Dado	At	Endereço	Bit	Comentário
DI1	0	BOOL	%IB	0	0	Entrada digital 1

Tag: identificação da variável
Tamanho: número de elementos do array* relacionado a variável
Tipo de Dado: tipo númerico da variável
At: define qual área de memória global a variável acessa
Endereço: endereço relacionado a área de memória global
Bit: para tipo de dados booleanos e necessário definir o bit que a mesma acessa (0...7)
Comentário: comentário da variável no idioma selecionado

* Quando tamanho > 0 as variáveis são acessadas no ladder através do seu índice de array

5.5 Editor Ladder/Rungs

O programa ladder é editado através de um editor gráfico que organiza o mesmo através de rungs. Cada rung do programa ladder corresponde a um intertravamento que relaciona elementos de entrada a elementos de saída. Esse editor é mostrado conforme figura a seguir.



Os elementos principais de uma rung são:
A - Barramento esquerdo
B - Barramento direito
C - Barramento de saídas*
D - Ligações entre elementos
E - Título e comentário da rung

* As saídas sempre estarão conectados a direita do barramento de saídas

Maiores detalhes da programação ladder clique aqui.

Através da barra de ferramentas do editor ladder/rungs é possível executar as seguintes operações:

- Adicionar uma nova rung
- Remover a rung selecionada
- Inserir um contato na rung
- Inserir um bloco funcional na rung
- Inserir um bobina de saída na rung
- Apagar um elemento da rung (contato, bloco funcional ou bobina)
- Selecionar a rung para edição

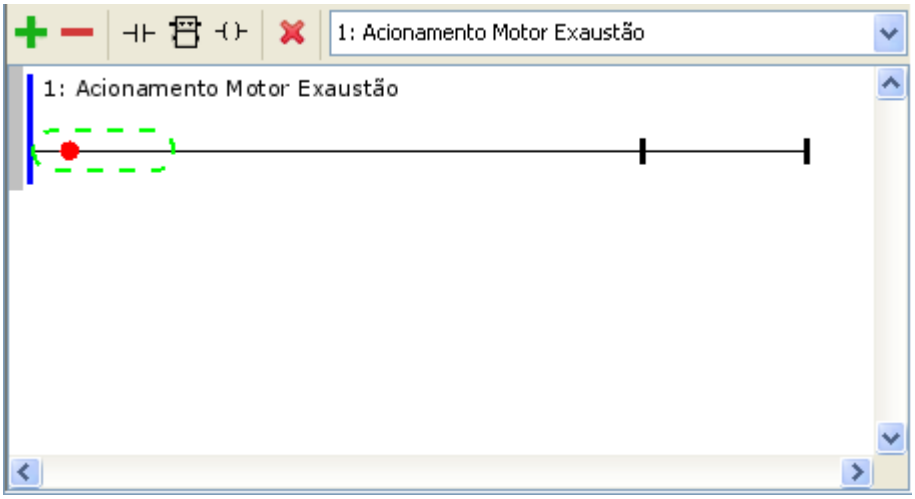
5.6 Editando uma Rung

Para editar um rung a mesma deve estar selecionada. Para fazer a seleção da rung você pode clicar com o mouse sobre a mesma ou utilizar o controle de seleção de rungs na barra de ferramentas do editor ladder. Sempre que a rung estiver selecionada existirá uma barra cinza na parte esquerda da rung conforme figura a seguir.

5.7 Editando Título e Comentário da Rung

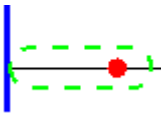
Para editar o título e comentário da rung basta dar um duplo clique do mouse sobre a região do título e comentário ou pressionar as teclas shift+F2. O seguinte diálogo aparecerá.

Após editar o título e comentário o editor ficará da seguinte maneira.



5.8 Inserindo Elementos na Rung

Para inserir um elemento ladder na rung é necessário que um elemento já existente do ladder esteja selecionado, conforme figura a seguir.

 - O elemento selecionado estará envolvido por um retângulo verde e riscado. Existirá também um ponto vermelho indicando aonde o novo elemento será inserido (ponto de inserção).

A inserção de elementos ladder pode ser feita através de três maneiras distintas.

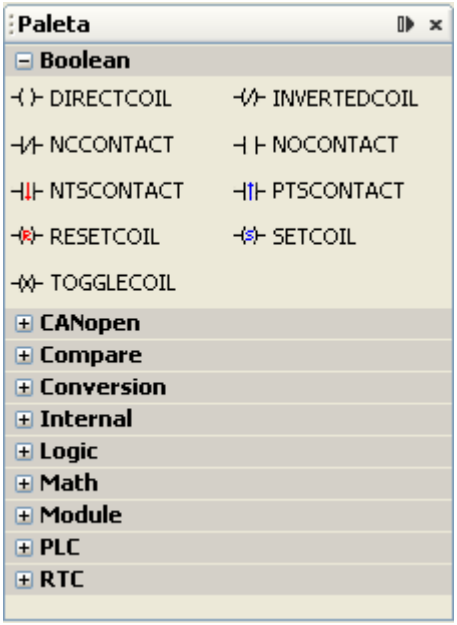
1 - Pelo teclado através do seguintes atalhos

tecla **C** - inserir contato
tecla **F** - inserir bloco funcional
tecla **O** - inserir bobina de saída

2 - Pela barra de ferramentas do editor ladder/rungs

- Inserir um contato na rung
- Inserir um bloco funcional na rung
- Inserir um bobina de saída na rung

3 - Pela paleta de elementos ladder*

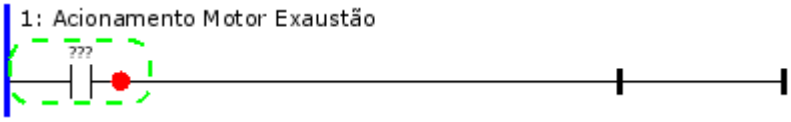


- Para inserir elementos pela paleta deve-se utilizar a operação arrastar e soltar do mouse. Para isso clique sobre o elemento da paleta, mantenha o mouse pressionado, mova o mouse até o ponto de inserção no ladder e

solte o botão do mouse.

* Caso não esteja visível ativar pelo menu Janela opção Paleta

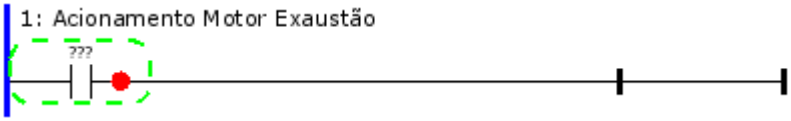
Após inserir um elemento, por exemplo um contato, a rung ficará da seguinte maneira.



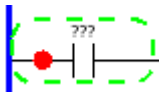
5.9 Navegação pelo Teclado na Rung

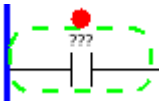
A navegação pelo teclado na rung é feito pelas teclas ←, →, ↑, ↓. Através dessas teclas e possível selecionar os diversos elementos que estão dentro da rung e também definir o ponto de inserção dos novos elementos.

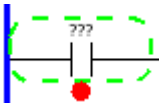
Conforme mostrado no item anterior ao inserir um elemento do tipo contato na rung a mesma ficará da seguinte maneira.

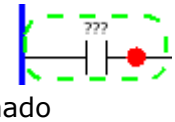


A pressionar as teclas de navegação podemos modificar o ponto de inserção de um novo elemento conforme figuras a seguir:

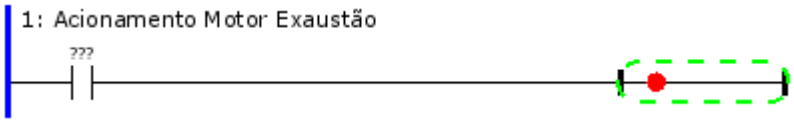
Tecla ←  indica que o novo elemento será inserido antes do elemento selecionado

Tecla ↑  indica que o novo elemento será inserido em paralelo acima do elemento selecionado

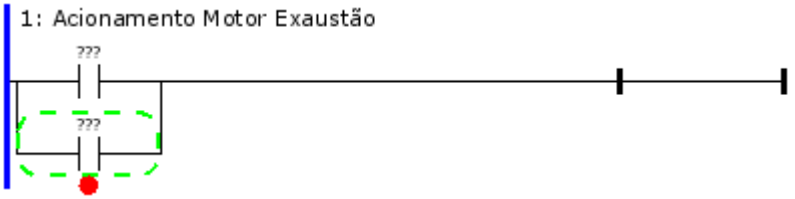
Tecla ↓  indica que o novo elemento será inserido em paralelo abaixo do elemento selecionado

Tecla →  indica que o novo elemento será inserido depois do elemento selecionado

Estando o ponto de inserção já selecionado e ao clicar a mesma tecla da direção correspondente a seleção passará para o próximo elemento conforme figura a seguir.

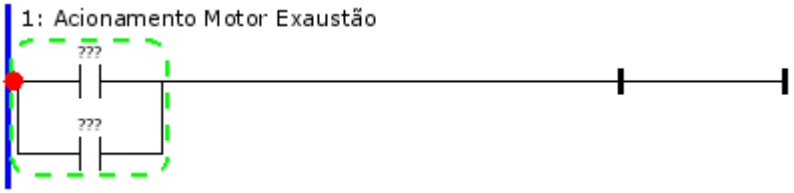


Ao inserir elementos em paralelo ao navegar pelos elementos navegamos por cada elemento individualmente conforme exemplo a seguir.

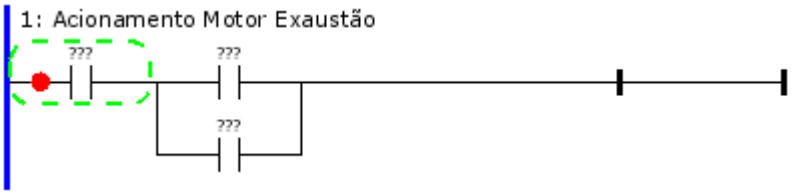


Nessa situação ao inserirmos um elemento estaremos fazendo essa inserção relacionada a esse elemento. Caso seja necessário inserir um elemento relacionado ao paralelo que é um grupo de elementos devemos utilizar a tecla **G** que selecionara o grupo e então poderemos definir o ponto de inserção e fazer a inserção conforme figuras a seguir.

- Pressionado a tecla **G:**

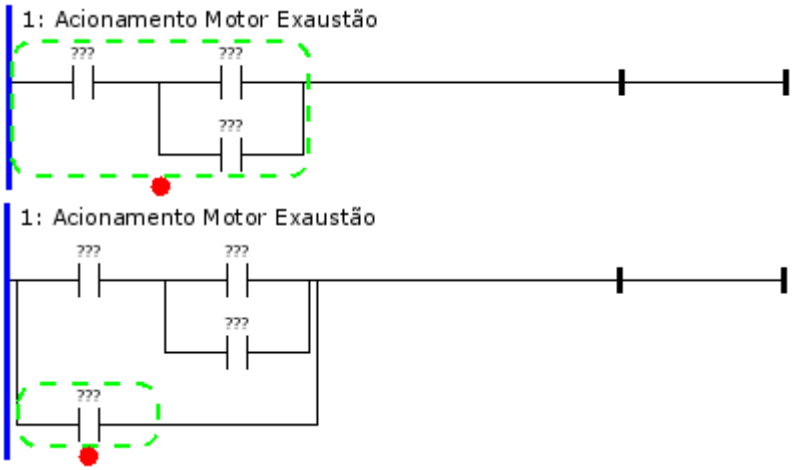


- Inserido contato antes do paralelo:

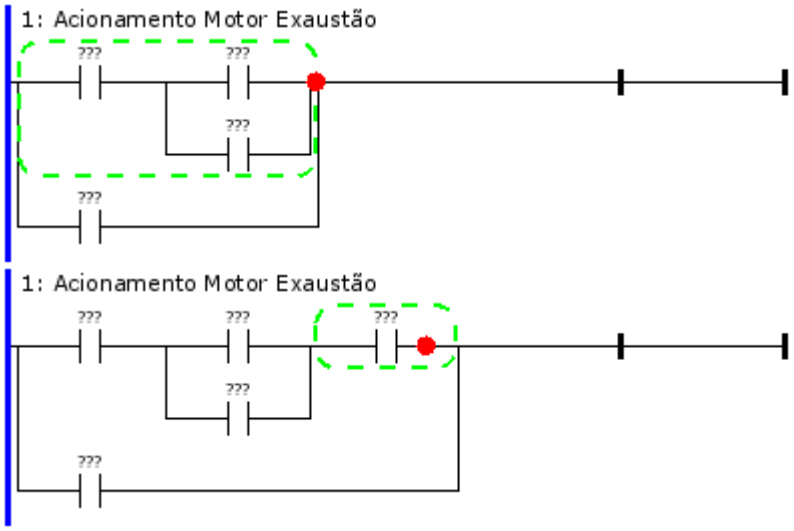


Ao cada toque da tecla **G** selecionamos o grupo imediatamente superior ao elemento/ grupo selecionado e ao final voltamos para o elemento original, então em algumas situações é necessário mais que um toque da tecla **G**. Através da tecla **G** é possível fazer diversas operações de inserção conforme exemplos a seguir.

- Inserir um elemento em paralelo ao grupo de elementos da rung

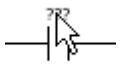


- Inserir um elemento em serie ao grupo formado por elementos internos a um paralelo

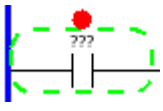


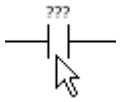
5.10 Navegação pelo Mouse na Rung

Todas as funcionalidades de navegação pelo teclado na rung também estão disponíveis no mouse. Diretamente pelo clique do mouse é possível selecionar um elemento e o respectivo ponto de inserção.

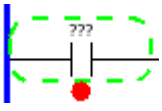


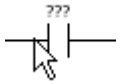
- Seleciona elemento e ponto de inserção superior

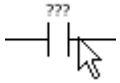
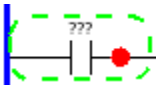




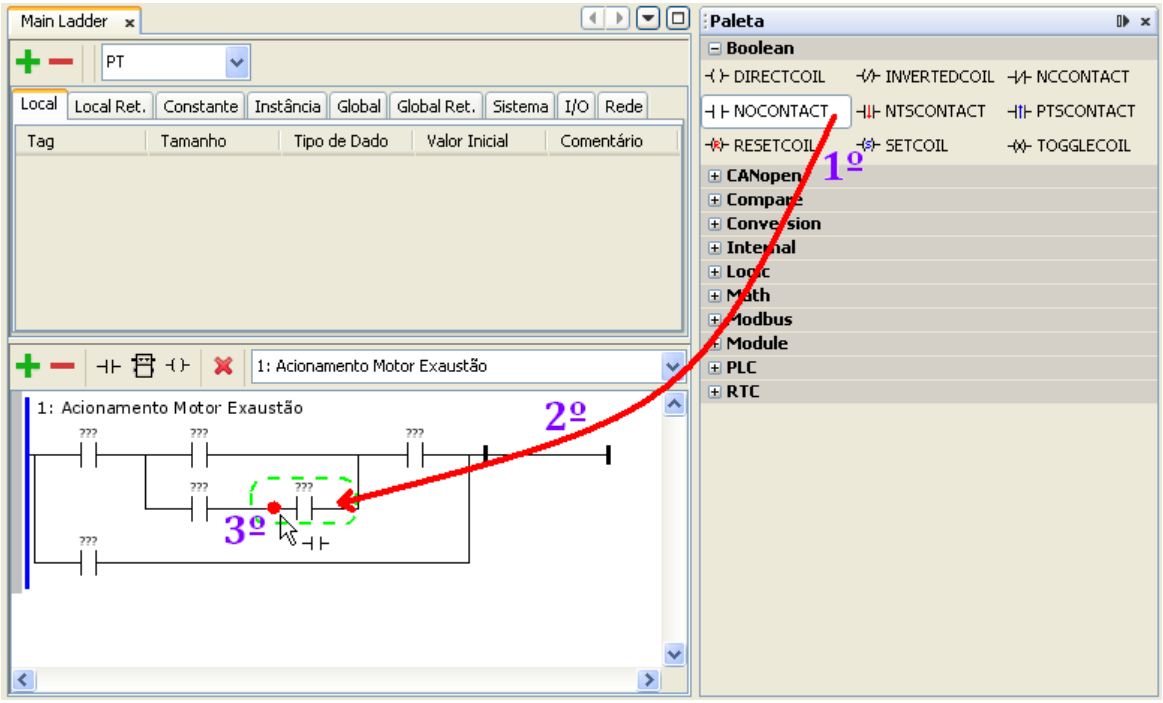
- Seleciona elemento e ponto de inserção inferior




- Seleciona elemento e ponto de inserção anterior

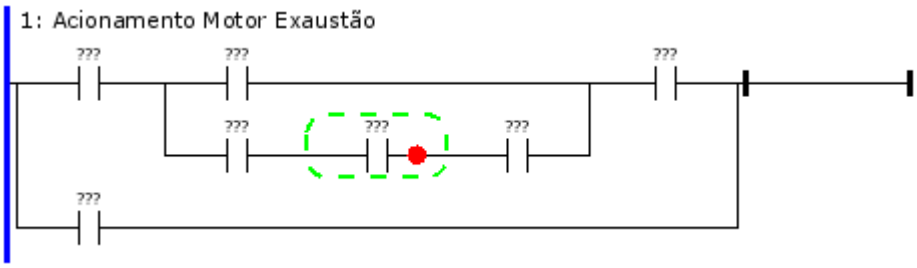

- Seleciona elemento e ponto de inserção posterior 

Durante a operação de inserção de elemento via mouse, através do arrastar e soltar do mouse da paleta, a medida que arrastamos os elementos em cima da rung a seleção e o ponto de inserção acompanha o cursor do mouse de modo a determinar o ponto de inserção. A figura a seguir exemplifica a inserção de elementos via mouse.



Sequência de inserção de elementos via mouse:
1 - Selecione o elemento da paleta que deseja inserir cliquando com o mouse sobre o mesmo e mantendo o botão do mouse pressionado
2 - Arraste esse elemento sobre a rung
3 - Selecione o ponto de inserção e solte o botão do mouse sobre ele

O resultado da inserção nesse exemplo será os seguinte.



Durante a inserção via mouse também é possível selecionar o grupo de elementos de maneira análoga a tecla G. Para isso durante o arraste do elemento previamente selecionado pressione e mantenha pressionada a tecla modificador Ctrl, enquanto a tecla Ctrl estiver pressionada o mouse ira selecionar pontos de inserção relacionados a grupos.

5.11 Inserindo Elementos em Série

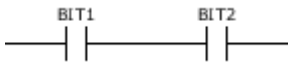
Através do ponto de inserção dos elementos é possível inserir um elementos em série. Para isso utilize os seguintes pontos de inserção:

- Insere elemento em série antes do elemento selecionado

- Insere elemento em série depois do elemento selecionado

Inserir elementos em série corresponde a lógica booleana AND conforme explicação a seguir:

LÓGICA AND



A figura acima executa uma Lógica AND entre os dois últimos elementos carregados no STACK, baixa um nível do STACK e joga o resultado para o topo do STACK. Isto significa que é executada a seguinte operação Booleana: topo do STACK = BIT1.BIT2.

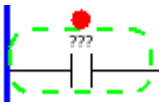
Em linguagem IL (Lista de Instruções) fica:
```
LD    BIT1    (* carrega o valor da variável BIT1 para o STACK *)
LD    BIT2    (* carrega o valor da variável BIT2 para o STACK *)
AND           (* Executa a Lógica AND entre o BIT1 e BIT2 através do STACK *)
```

Tabela Verdade

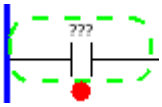
BIT1	BIT2	STACK
0	0	0
0	1	0
1	0	0
1	1	1

5.12 Inserindo Elementos em Paralelo

Através do ponto de inserção dos elementos é possível inserir um elementos em paralelo. Para isso utilize os seguintes pontos de inserção:



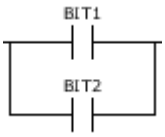
- Insere elemento em paralelo acima do elemento selecionado



- Insere elemento em paralelo abaixo do elemento selecionado

Inserir elementos em série corresponde a lógica booleana OR conforme explicação a seguir:

LÓGICA OR



A figura acima executa uma Lógica OR entre os dois últimos elementos carregados no STACK, baixa um nível do STACK e joga o resultado para o topo do STACK. Isto significa que é executada a seguinte operação Booleana: topo do STACK = BIT1 + BIT2.

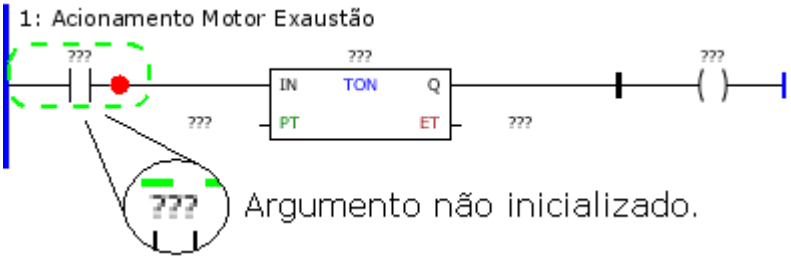
Em linguagem IL (Lista de Instruções) fica:
```
LD    BIT1   (* carrega o valor da variável BIT1 para o STACK *)
LD    BIT2   (* carrega o valor da variável BIT2 para o STACK *)
OR           (* Executa a Lógica OR entre o BIT1 e BIT2 através do STACK *)
```

Tabela Verdade

BIT1	BIT2	STACK
0	0	0
0	1	1
1	0	1
1	1	1

5.13 Edição de Variáveis na Rung

Os elementos ladder inseridos na rung necessitam que variáveis sejam definidas para cada argumento. Por padrão todos argumentos não são inicializados e necessitam da edição pelo usuário. Veja figura a seguir.

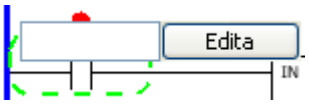


Para definir uma variável para o argumento é necessário entrar no modo de edição do argumento que é feito através de duas maneiras.

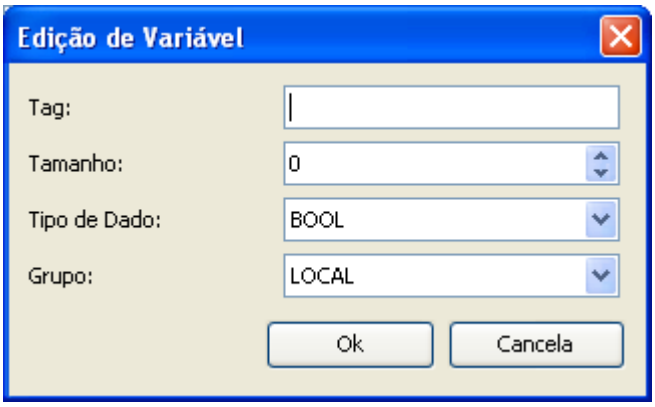
1 - Pelo mouse:
Executando um duplo clique do mouse diretamente no argumento

2 - Pelo teclado:
Pressionando a tecla F2. Para elementos com 1 argumento entra no modo de edição de argumento diretamente, nos outros elementos com mais de 1 argumento e necessário selecionar o argumento através das teclas direcionais e então pressionar novamente a tecla F2.

Ao entrar no modo de edição o elemento aparecerá similar a seguinte figura.



Neste momento uma caixa de edição estará habilitada para que você entre com o nome da variável. Ao pressionar o botão "Edita" uma caixa para criar a nova variável será habilita.

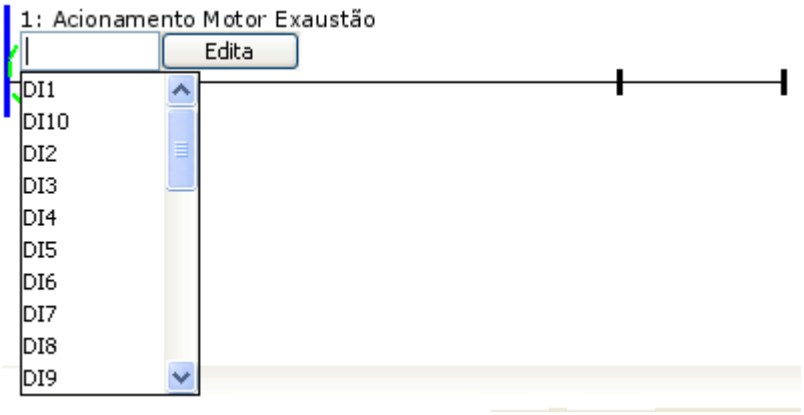


Nesta caixa você deverá definir as seguintes opções para a variável:

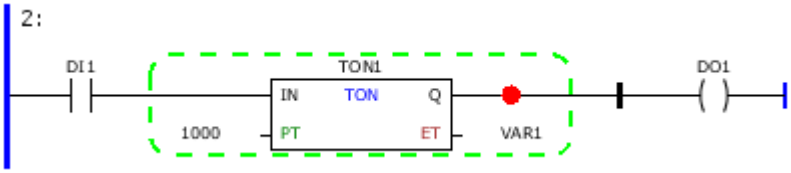
Tag: identificação da variável
Tamanho: número de elementos do array* relacionado a variável
Tipo de Dado: tipo numérico da variável
Grupo: Grupo a qual a variável pertence

Caso já existam variáveis definidas para o tipo compatível com o elemento ladder uma caixa de seleção com essas variáveis aparecerá em conjunto com a caixa de

edição. Para selecionar a variável desejada aperte a tecla direcional para baixo e após estar com a variável selecionada pressione enter. A figura a seguir mostra essa funcionalidade.



Nos blocos funcionais também é possível a entrada de valores literais conforme figura a seguir.



Nesse exemplo foi configurado a entra PT do bloco TON com o valor 1000 que é um literal.

Para a entrada de literais deve ser usado a seguintes convenções:

- Número inteiro não tem ponto.
  Ex. 12, 1000, 1555

- Número reais em ponto flutuante devem obrigatoriamente ter ponto.
  Ex. 1.5, 2.25, 3.0

- Números representados em hexadecimal obrigatoriamente devem definir o tipo do dado.
  Ex. BYTE#16#7F, WORD#16#3CF0, DWORD#16#00FF0088

- Números representados em binário obrigatoriamente devem definir o tipo do dado.
  Ex. BYTE#2#1010_0000, WORD#2#0111_0000_0000_0001

6 Tarefas

6.1 Configuração de Tarefas

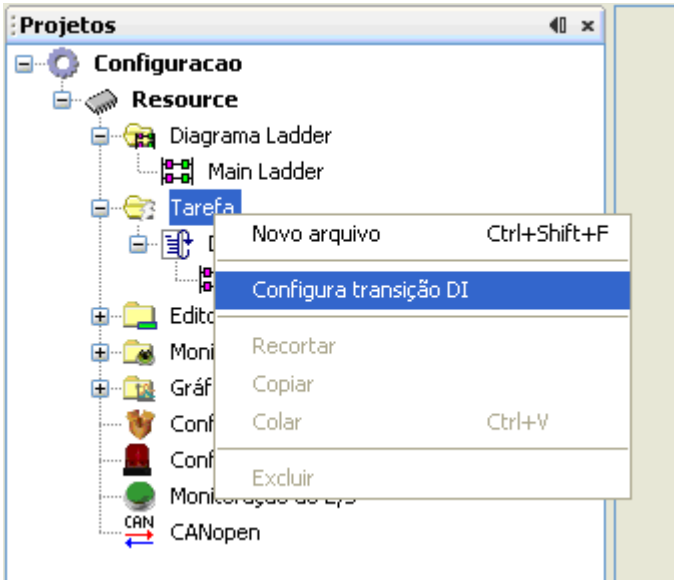
Visão Geral

A tarefa é um mecanismo de escalonamento muito útil em sistemas de tempo real para controlar a execução de programas, de forma periódica ou disparada por eventos.

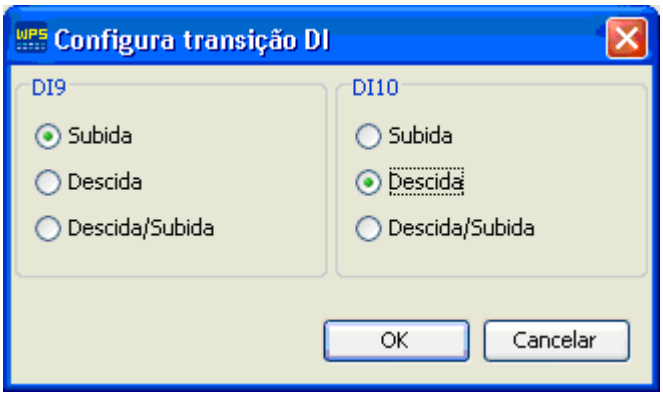
A configuração das tarefas é realizada adicionando arquivos de tarefas à pasta "Tarefa", dentro do recurso. A estes arquivos de tarefa, são então associados programas que executam sua lógica, quando a tarefa se torna ativa. Para cada tarefa pode ser configurado um watchdog (ver seção watchdog). Cada tarefa possui um marcador de sistema associado que pode ser habilitado ou desabilitado através do programa.

Dentro de um recurso, existe uma única tarefa chamada "Main", que não pode ser apagada, na qual o programa principal (Main Ladder) é executado.

No menu de opções da pasta de tarefas, é possível realizar as seguintes ações:

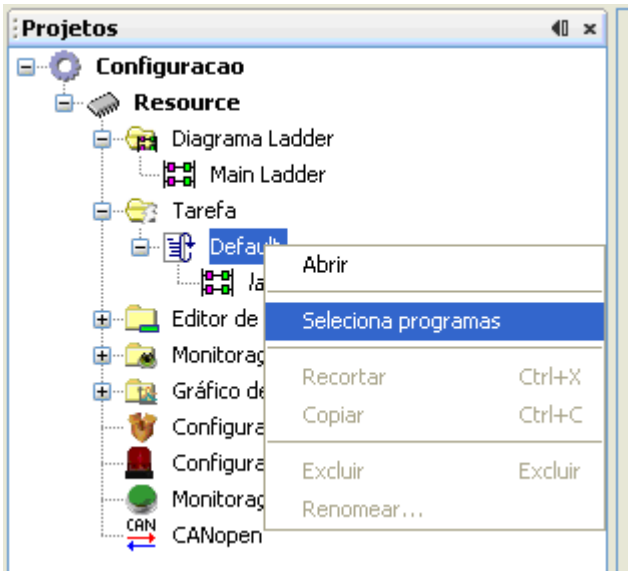


Adicionar uma nova tarefa: para adicionar uma nova tarefa, é preciso selecionar a opção "Novo arquivo" da pasta "Tarefa".



Estas DIs são utilizadas nas tarefas do tipo evento externo e contagem.

Através do menu de opções dos arquivos das tarefas, é possível realizar as seguintes ações:



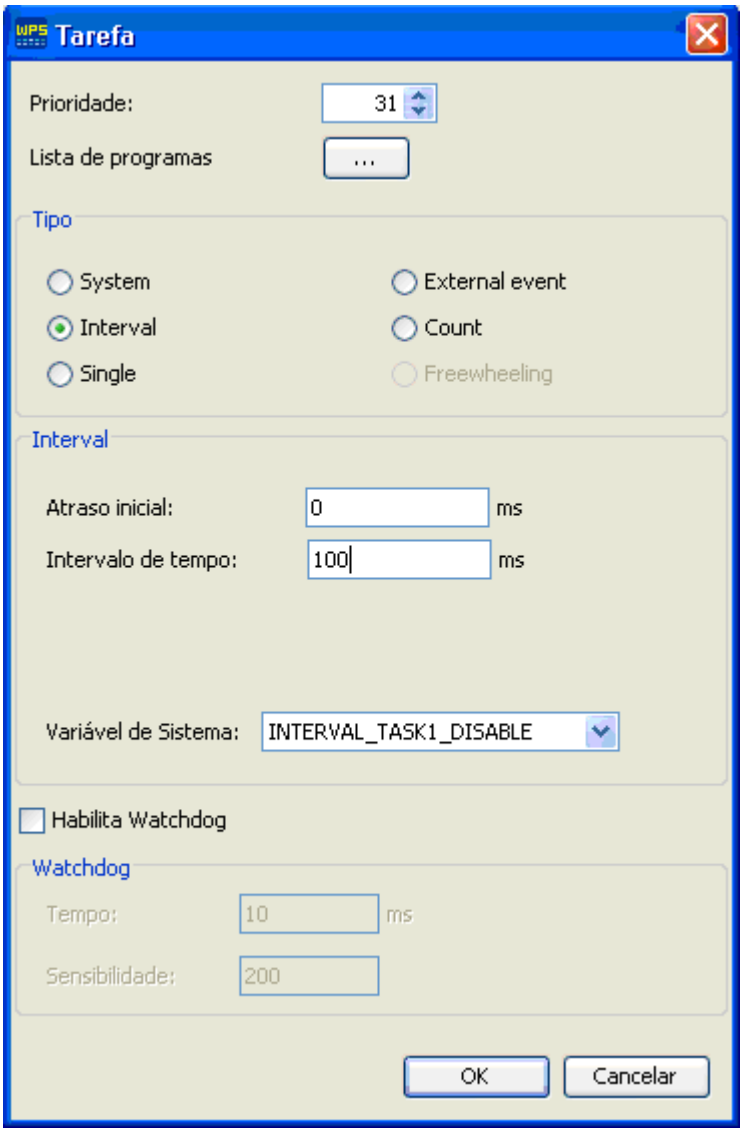
Configurar tarefas: para configurar a tarefa, é preciso selecionar a opção "Abrir", no arquivo de tarefas que se deseja configurar. Para mais informações de como configurar as tarefas, ver seção "Configuração das tarefas".

Adicionar, remover ou ordenar programas da tarefa: para adicionar, remover ou ordenar programas de uma determinada tarefa, é preciso selecionar a opção "Seleciona programas" no arquivo de tarefas que deseja configurar. Para mais informações de como selecionar os programas, ver na seção "Configuração das tarefas" o item "Lista de programas".

Configuração da borda de transição das entradas digitais (DIs): as entradas digitais DI9 e DI10 possuem a opção de disparar eventos através da transição da borda de subida, descida ou subida e descida. A configuração destas bordas de transição é feita através da janela de configuração das DIs, que é acessada selecionando a opção "Configura transição DI" da pasta de "Tarefa".

Configuração de Tarefas

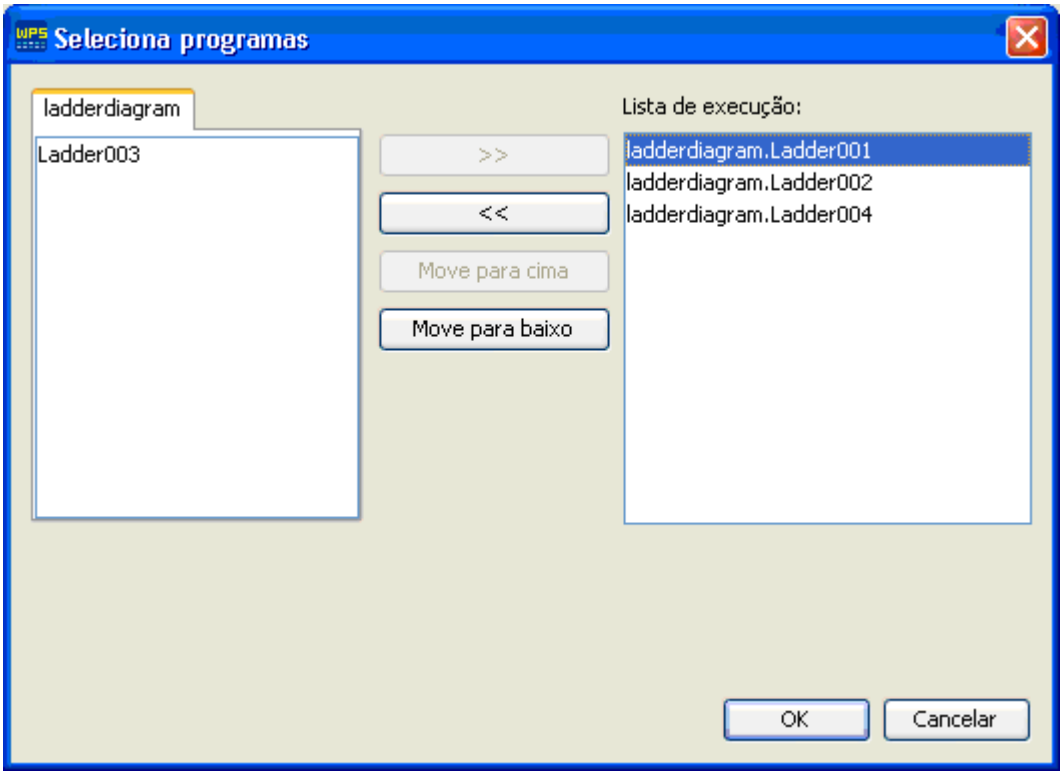
Na janela de configuração das tarefas, são configurados:
- Prioridade,
- Sequência dos programas,
- Tipo de tarefa, e
- Opções de watchdog da tarefa.



Prioridade: configurada com um número entre 0 e 30 (0 - prioridade mais alta, 30 - mais baixa) que define a prioridade em que a tarefa será executada. Caso uma tarefa de prioridade mais alta do que a tarefa em execução seja acionada, esta imediatamente entrará em execução. De outra forma, caso uma tarefa de prioridade mais baixa do que a atual seja acionada, esta aguarda o final da execução da tarefa atual para entrar em execução.

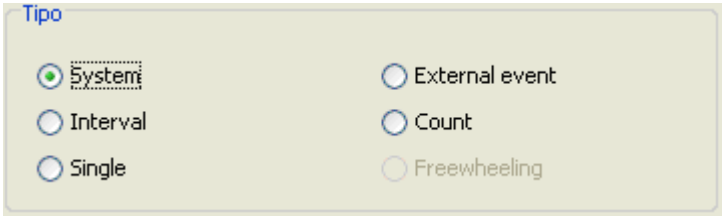
Lista de programas: a janela de seleção da sequência dos programas serve para adicionar, remover ou alterar a sequência dos programas associados a determinada

tarefa. Na lista do lado esquerdo, estão todos os programas disponíveis no recurso e no lado direito, os programas selecionados para esta tarefa. A ordem que os programas entrarão em execução será a mesma ordem definida nesta lista.



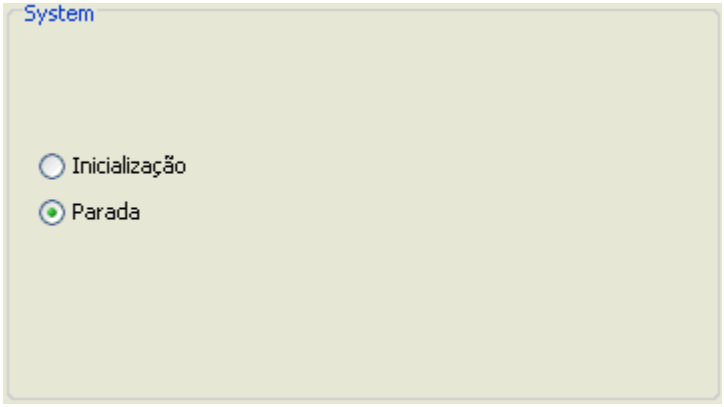
Tipos de Tarefa

As tarefas estão dividas em: system, interval, single, external event, count e freewheeling.



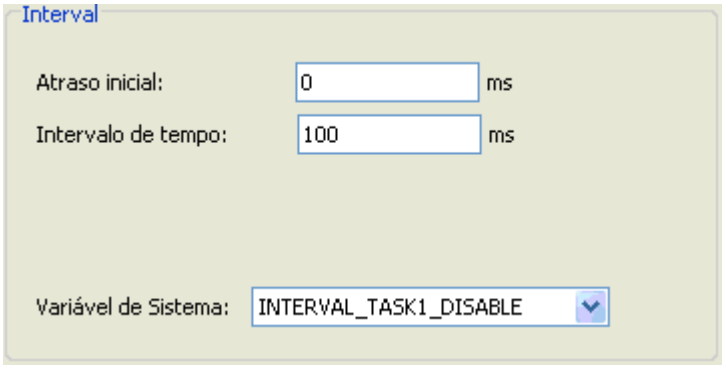
System

Esta tarefa pode ser do tipo Inicialização ou Parada. Quando selecionada a opção "Inicialização", a tarefa entrará em execução logo que o programa do usuário for executado pela primeira vez. Caso a opção "Parada" seja selecionada, os programas associados a esta tarefa, entram em execução logo após a parada do programa do usuário.



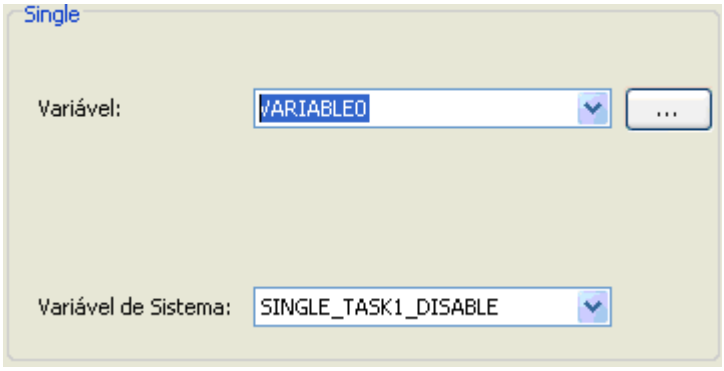
Interval

Os programas associados a esta tarefa, são executados repetidas vezes, no intervalo de tempo definido no campo "Intervalo de tempo", com o tempo de atraso para a primeira execução definido no campo "Atraso inicial". A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.



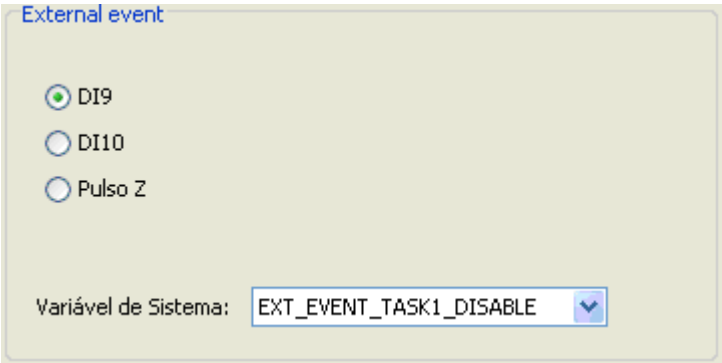
Single

Os programas associados a esta tarefa, são executados uma única vez, sempre que a variável selecionada sofrer uma transição positiva, ou seja, de zero para algum outro valor. A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.



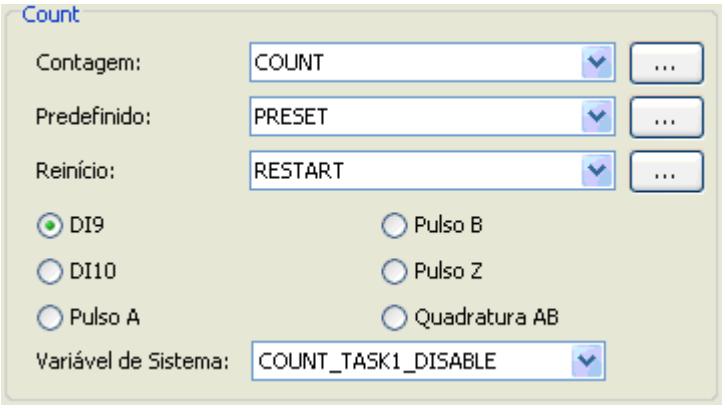
External Event

Os programas associados a esta tarefa, são executados uma única vez, sempre que o evento DI9, DI10 ou Pulso Z selecionado for ativado. Os eventos DI9 e DI10 possuem a opção de selecionar a transição que ativa a tarefa (ver item "Configuração da borda de transição das entradas digitais" na seção "Visão Geral"). A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.



Count

Os programas associados a esta tarefa, são executados toda vez que a contagem de pulsos na entrada selecionada (DI9, DI10, Pulso A, Pulso B, Pulso Z e Quadratura AB), ultrapassar o valor da variável definida no campo "Predefinido". O valor de contagem dos pulsos é armazenado na variável "Contagem". O valor da contagem dos pulsos é reiniciado quando este valor ultrapassar o valor definido no campo "Reinício". As variáveis configuradas nestes campos podem ser do tipo DWORD e UDINT para as entradas DI9, DI10, Pulso A, Pulso B e Pulso Z e DINT para entrada quadratura AB. A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.

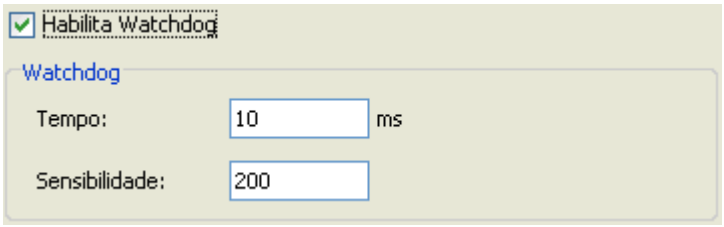


Freewheeling

Os programas associados a esta tarefa, são executados de forma cíclica. No final da execução da lista de programas, a lista de programas é reiniciada, até que o programa do usuário seja parado. O programa principal (Main Ladder) está associado a esta tarefa e não é possível removê-lo ou associá-lo a outra tarefa.

Watchdog

Quando a opção watchdog está ativa, o programa do usuário é parado com alarme de erro, caso o tempo de execução da tarefa demore mais do que o tempo definido no campo "Tempo", por um número de vezes maior que o definido no campo "Sensibilidade".

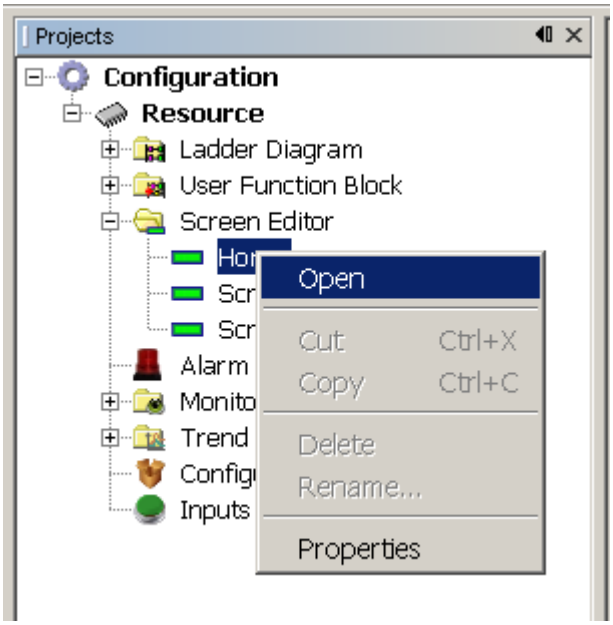


Tempo: Tempo máximo para execução da tarefa, sem que ocorra o erro de watchdog.
Sensibilidade: Número de erros de watchdog permitidos, antes da geração do alarme de erro e parada do programa do usuário.

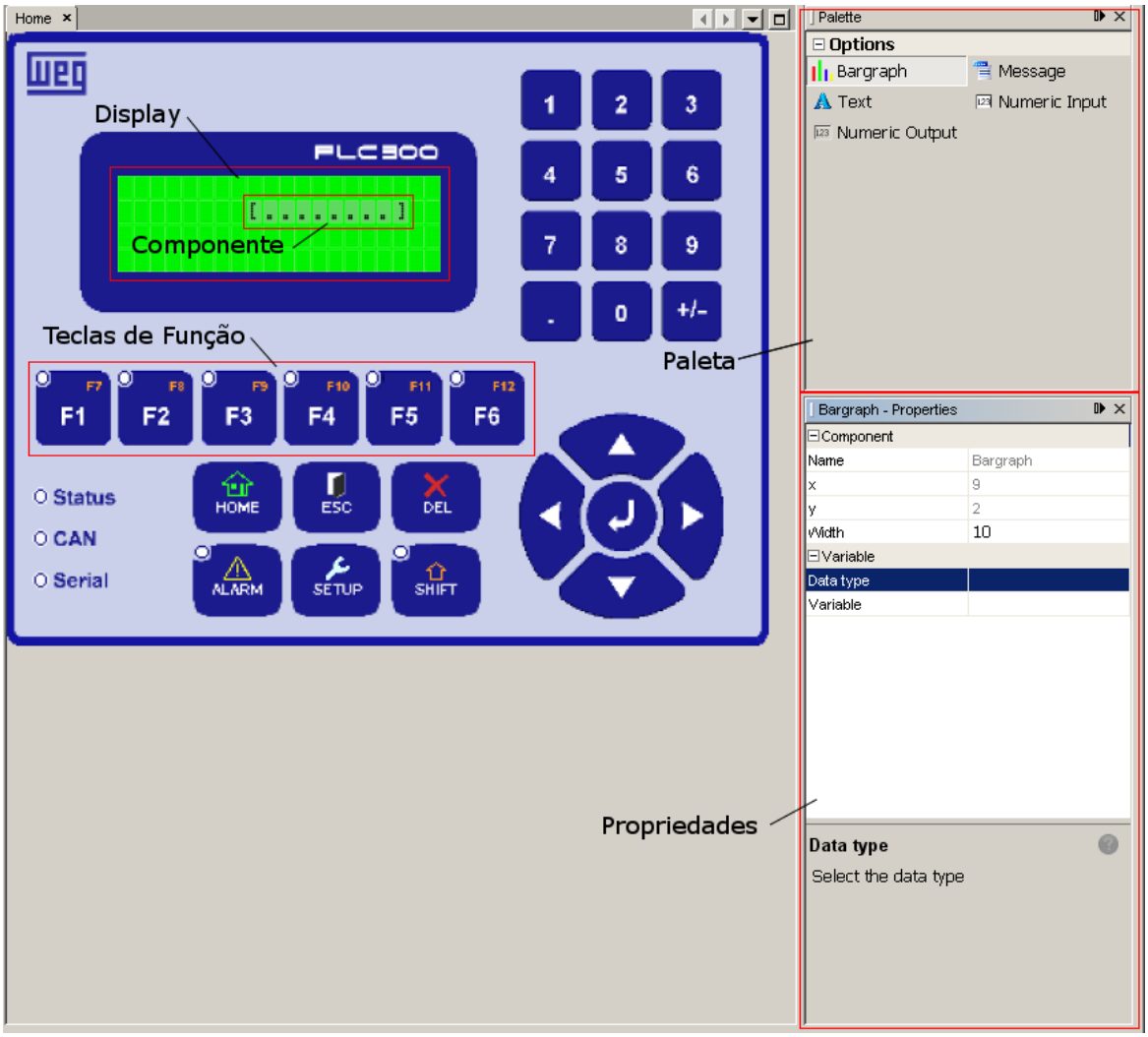
7 Editor de Telas

7.1 Configuração das Telas

O PLC 300 permite que até 240 telas possam ser programadas pelo usuário. Para iniciar o modo de edição das telas é preciso abrir uma tela na janela de projetos.



Após a abertura, a janela de edição das telas é visualizada. Esta janela é dividida nas seguintes áreas:



Componente
Componente configurável contendo determinada característica para composição da tela do equipamento.

Display
Local onde são inseridos e editados os componentes.

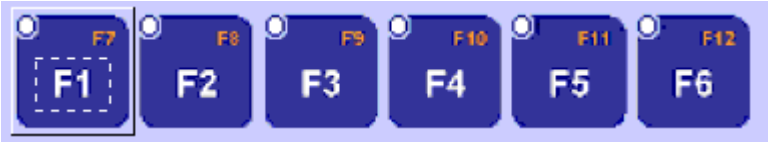
Teclas de Função
Teclas programáveis com funções de gerenciamento de telas e manipulação de bits.

Paleta
Janela contendo os componentes da tela do equipamento inseridos através de drag'n drop.

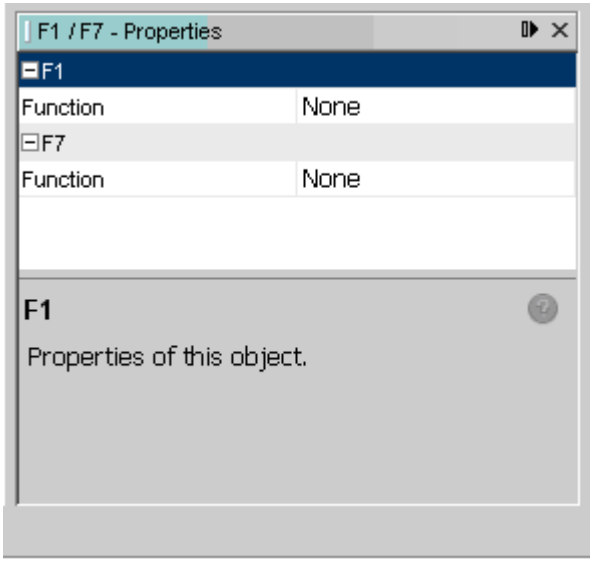
Janela de Propriedades
Janela sensível ao contexto onde são editadas e visualizadas as propriedades dos componentes e do display.

7.2 Teclas de Função

O gerenciamento de telas e a manipulação de bits são feitos através da configuração das teclas de função. Para tornar as propriedades de uma das teclas de função ativa é preciso selecioná-la clicando sobre ela na área das teclas de função.



As propriedades da tecla de função se tornarão ativas na tela de propriedades.

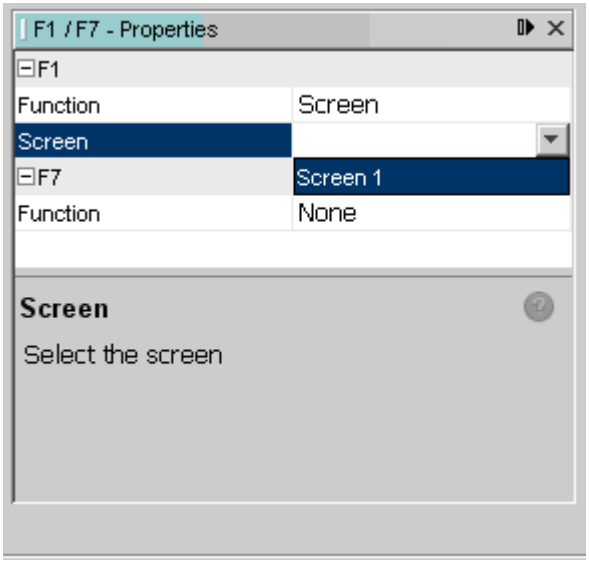


7.3 Gerenciamento de Telas

O editor de telas possui a função de gerenciamento de telas para o acesso navegação entre diferentes telas através das teclas de função.

Nota! Para esta configuração é necessário que uma tecla de função esteja selecionada e a janela de propriedades esteja aberta no ambiente de desenvolvimento.

A ligação entre as telas é feita através da edição da propriedade "Função" selecionando a opção "Screen". Após a seleção, uma lista com as telas disponíveis pode ser selecionada na opção "Tela".



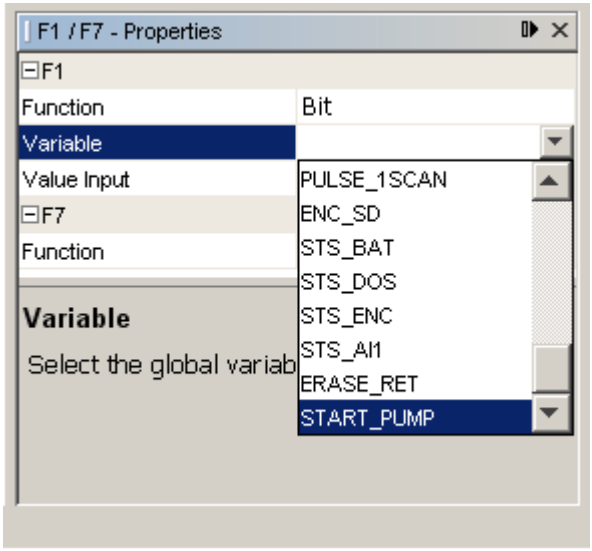
Neste exemplo, a tela selecionada (Screen 1) será acessada através da tecla de função configurada (F1) pela tela atual.

7.4 Manipulação de Bits

O editor de telas possui a função de manipulação de bits para alteração do estado de variáveis globais no ladder através das teclas de função.

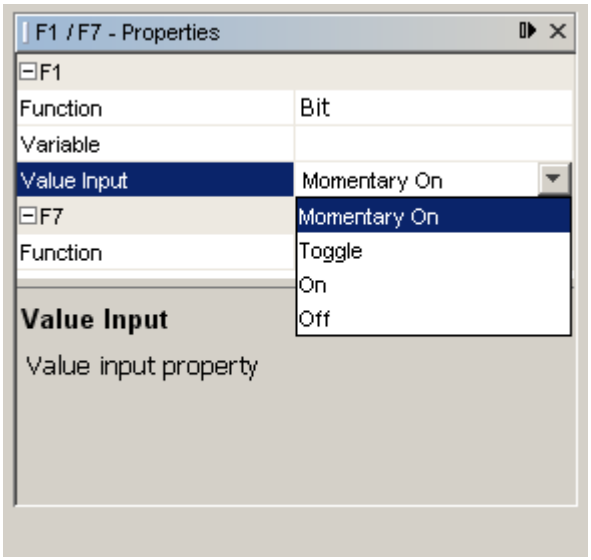
> Nota! Para esta configuração é necessário que uma tecla de função esteja selecionada e a janela de propriedades esteja aberta no ambiente de desenvolvimento.

Na manipulação de bits é necessária a edição da propriedade "Função" selecionando a opção "Bit".
Após a seleção da função "Bit" uma variável global deve ser selecionada na propriedade "Variável".



Nota! A variável deve ser previamente configurada no editor ladder.

As ações disponíveis para a função BIT são:



Momentary On
Liga uma determinada variável do tipo booleana (marcador de bit) somente enquanto a tecla estiver pressionada. Ao soltar a tecla, o marcador volta a zero.

Toggle
Inverte o estado de uma determinada variável do tipo booleana (marcador de bit);

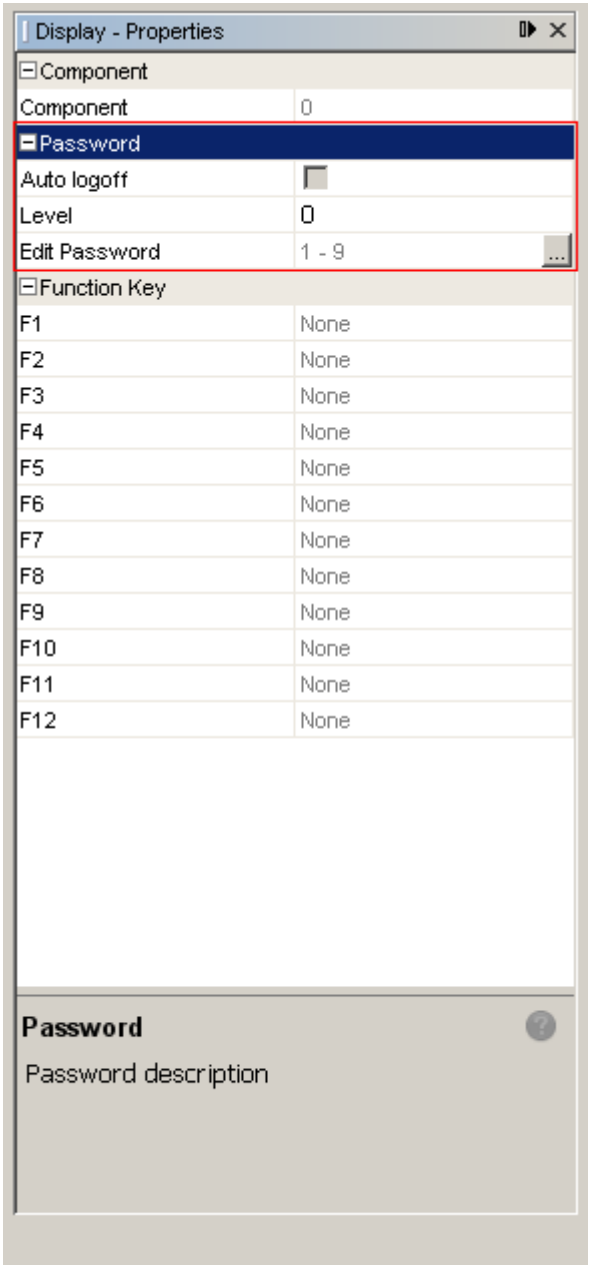
On
Liga uma determinada variável do tipo booleana (marcador de bit);

Off

Desliga uma determinada variável do tipo booleana (marcador de bit);

7.5 Senhas

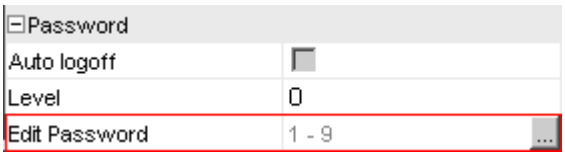
O PLC300 possui dez níveis de senha (0 - 9). Destes dez níveis, o nível zero é escolhido quando se deseja o acesso à tela por qualquer usuário. Com exceção da tela zero (HOME), que possui nível de acesso zero sem possibilidade de alteração, as demais telas podem ser programadas com um nível de acesso. As senhas configuradas devem possuir seis dígitos decimais. As propriedades para configuração dos níveis de acesso são acessadas através da janela de propriedades do display.



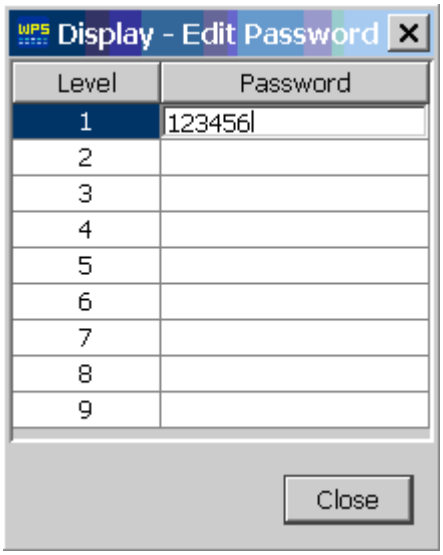
Nota! Para esta configuração é necessário que a janela de propriedades esteja aberta no ambiente de desenvolvimento.

As propriedades de configuração das senhas são as seguintes:

Editar Senha
Configura as senhas de 1 - 9 de acesso às telas. Para edição das senhas deve ser selecionado o botão ... na propriedade "Editar Senha".

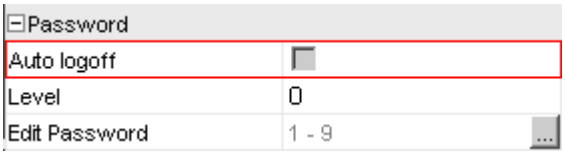


Então é aberta uma janela com uma tabela contendo campos para serem preenchidos com a senha.



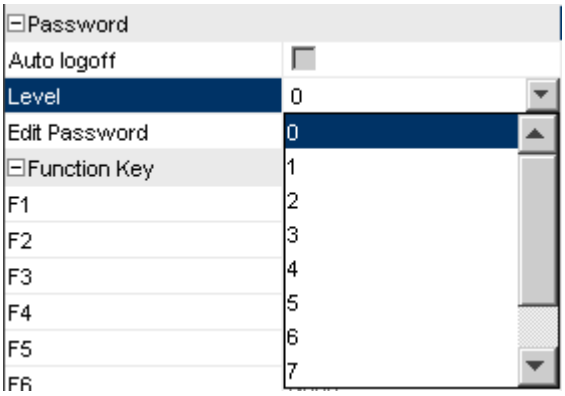
Após o final da edição basta fechar a janela que as senhas estarão gravadas.

Auto Logoff
Desativa a senha de acesso digitada para o nível anterior ao da tela que acabou de sair.



Nível
Seleciona o nível de acesso da tela. Quanto maior o nível de acesso, maior o privilégio. Até nove níveis de acesso podem ser programados, de modo a permitir diferentes tipos de usuários com acesso a determinadas telas. O nível de acesso zero

significa que a tela é de livre acesso.



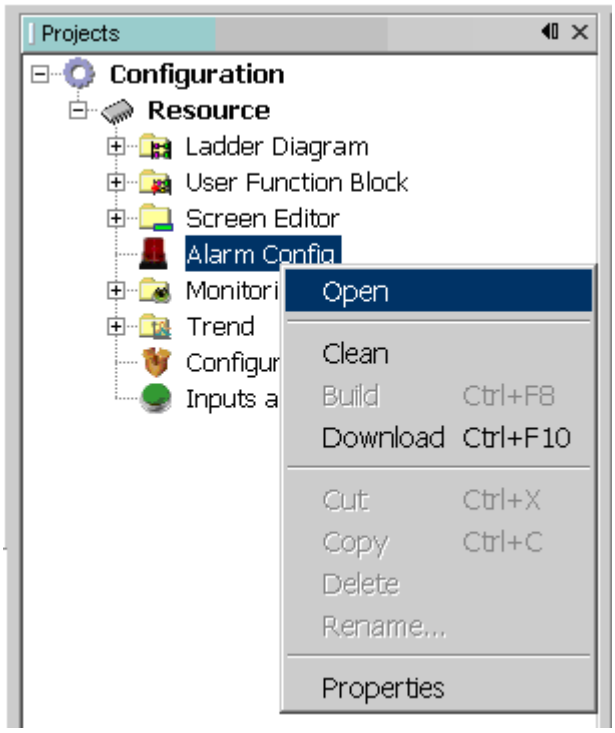
7.6 Alarmes

Os alarmes são uma ferramenta importante na automação de processos, permitindo ao usuário monitorar sua planta, checando pontos críticos e sinalizando ao operador.

No PLC300 os alarmes configuráveis são programados pelo usuário, sendo acionados por um marcador de bit que pode ser ativado pelo programa ladder.

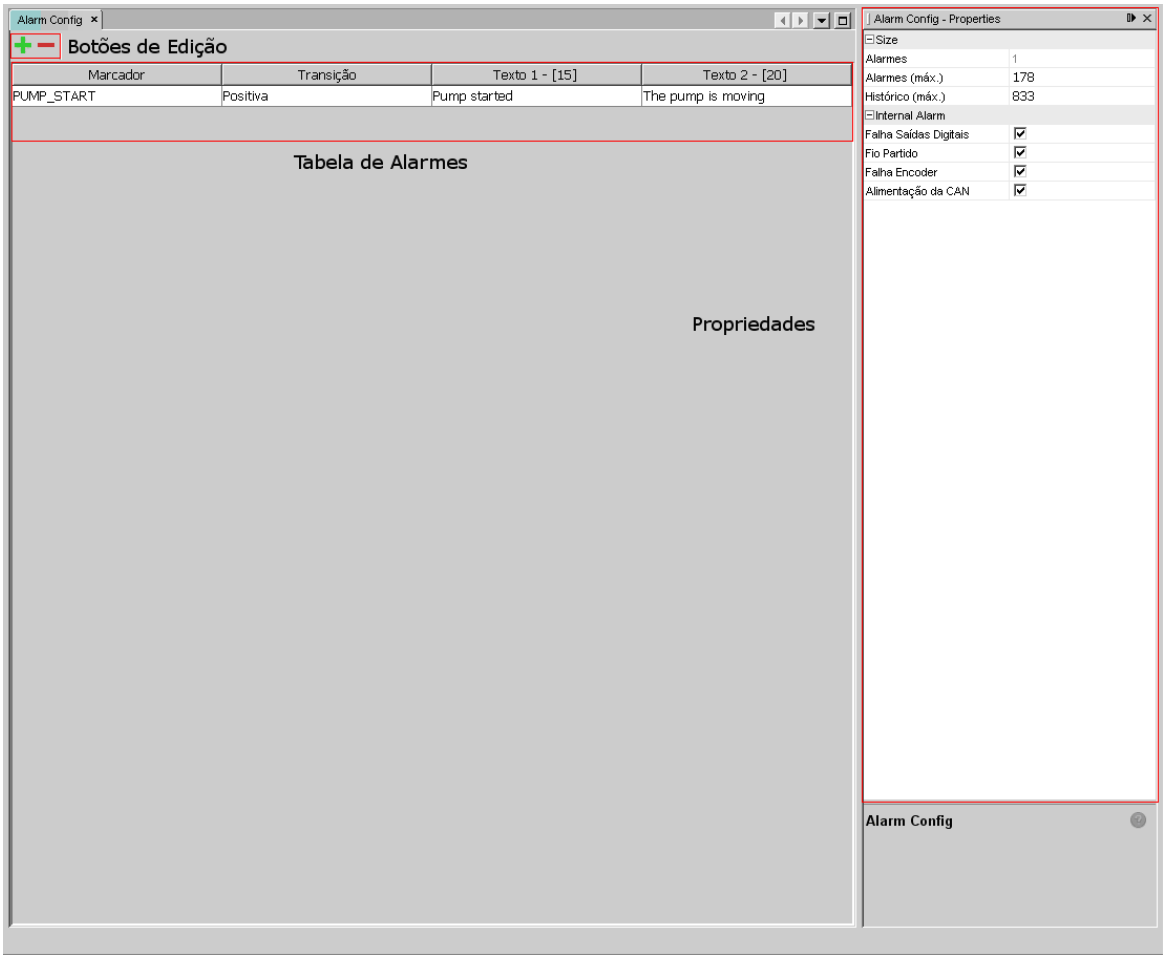
Os alarmes internos são alarmes relativos a alguns componentes de hardware, que ocupam a memória interna.

Os alarmes são configurados através da tela "Alarm Config" acessada pelas pastas de projeto:


Projects
Configuration
Resource
Ladder Diagram
User Function Block
Screen Editor
Alarm Config
Monitori
Trend
Configur
Inputs a
Open
Clean
Build Ctrl+F8
Download Ctrl+F10
Cut Ctrl+X
Copy Ctrl+C
Delete
Rename...
Properties

7.7 Alarmes Configuráveis

Os alarmes configuráveis são configurados através da janela de edição dos alarmes.



Os botões de edição dos alarmes são os seguintes:

Adiciona um novo alarme na tabela.

Remove um alarme da tabela.

A tabela dos alarmes possui quatro campos a serem preenchidos:

Marcador	Transição	Texto 1 - [15]	Texto 2 - [20]
PUMP_START	Positiva	Pump started	The pump is moving

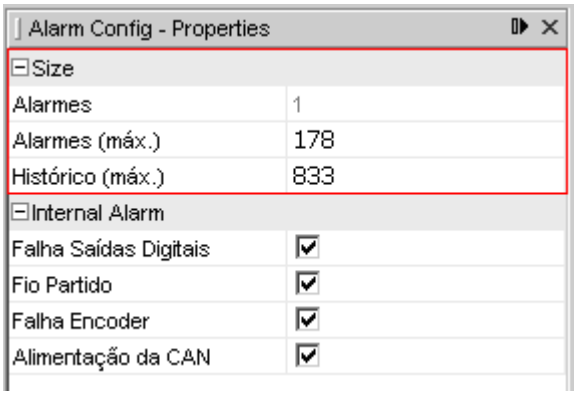
Marcador
Marcador de bit que ativa o alarme. Variável global do tipo booleana.

Transição
Borda de transição em que o alarme será ativado. Os valores possíveis são positiva (de 0 para 1) ou negativa (de 1 para 0).

Texto 1
Texto para com a mensagem de alarme. Este campo poderá conter no máximo 15 caracteres.

Texto 2
Texto com o descritivo da mensagem de alarme. Este campo poderá conter no máximo 20 caracteres.

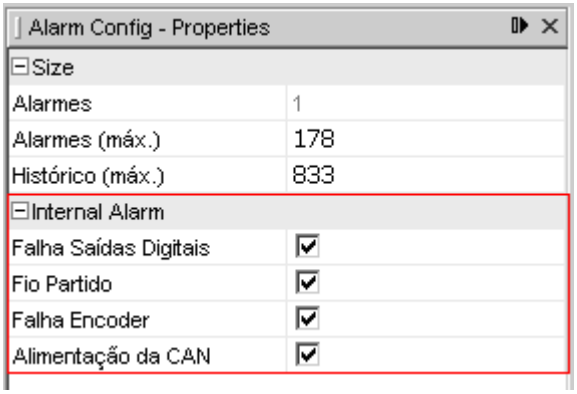
Através da janela de propriedades dos alarmes, na categoria "Tamanho", na opção "Alarmes (máx)" e "Histórico (máx)", é possível configurar a área reservada na memória para os alarmes configuráveis e histórico de alarmes. O valor informado corresponde ao número de máximo de alarmes permitido:



Nota! Para esta configuração é necessário que a janela de propriedades esteja aberta no ambiente de desenvolvimento.

7.8 Alarmes Internos

Os alarmes internos são configurados através da janela de propriedades dos alarmes na categoria alarmes internos:



O PLC 300 possui quatro alarmes internos que podem ser habilitados pelo usuário:

Falha Saídas Digitais
Indica que algum problema está ocorrendo em alguma das saídas DO1 a DO8.

Fio Partido
Corrente abaixo de 2 mA, quando a entrada analógica estiver em modo corrente 4 a 20mA.

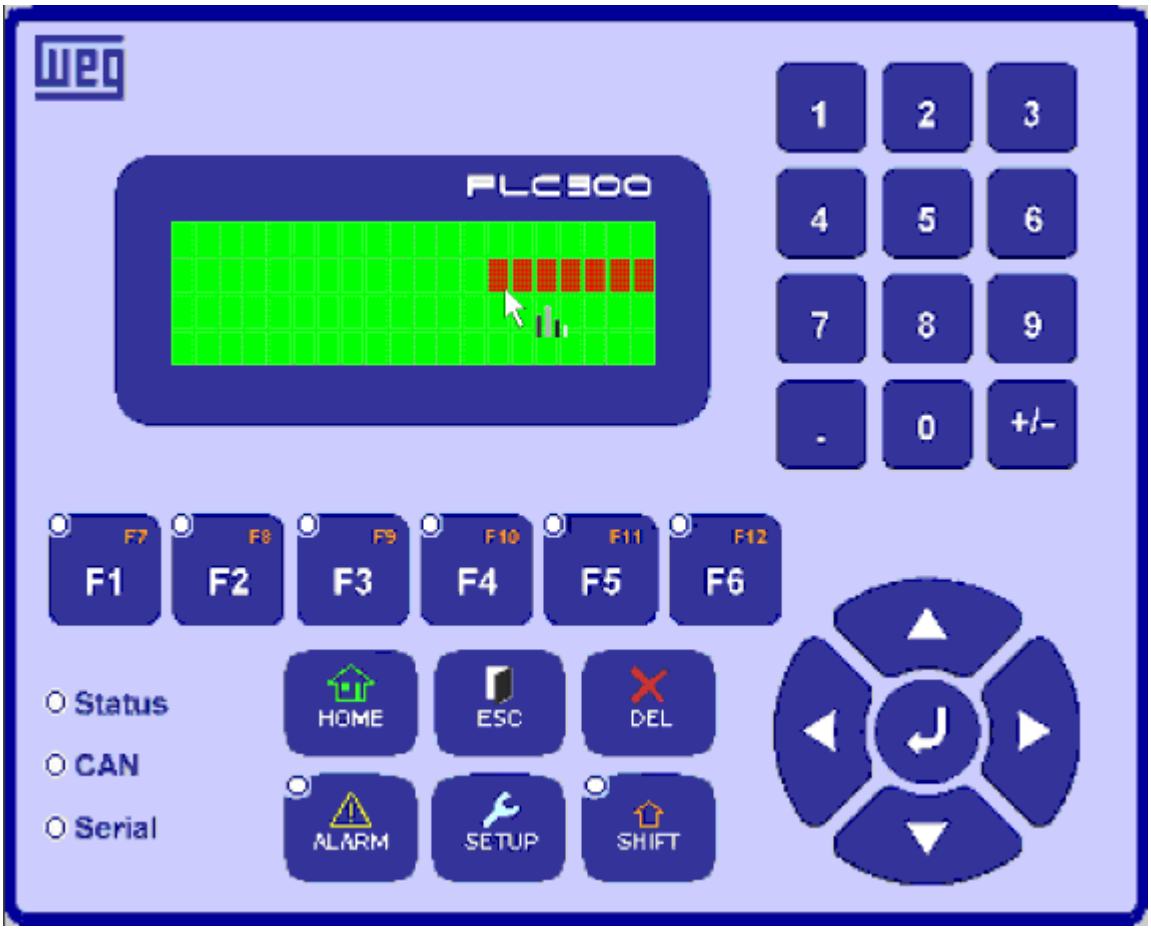
Falha Encoder
Falta de um dos sinais do encoder.

Alimentação da CAN
Ausência da tensão de alimentação da interface CAN.

7.9 Programação das Telas

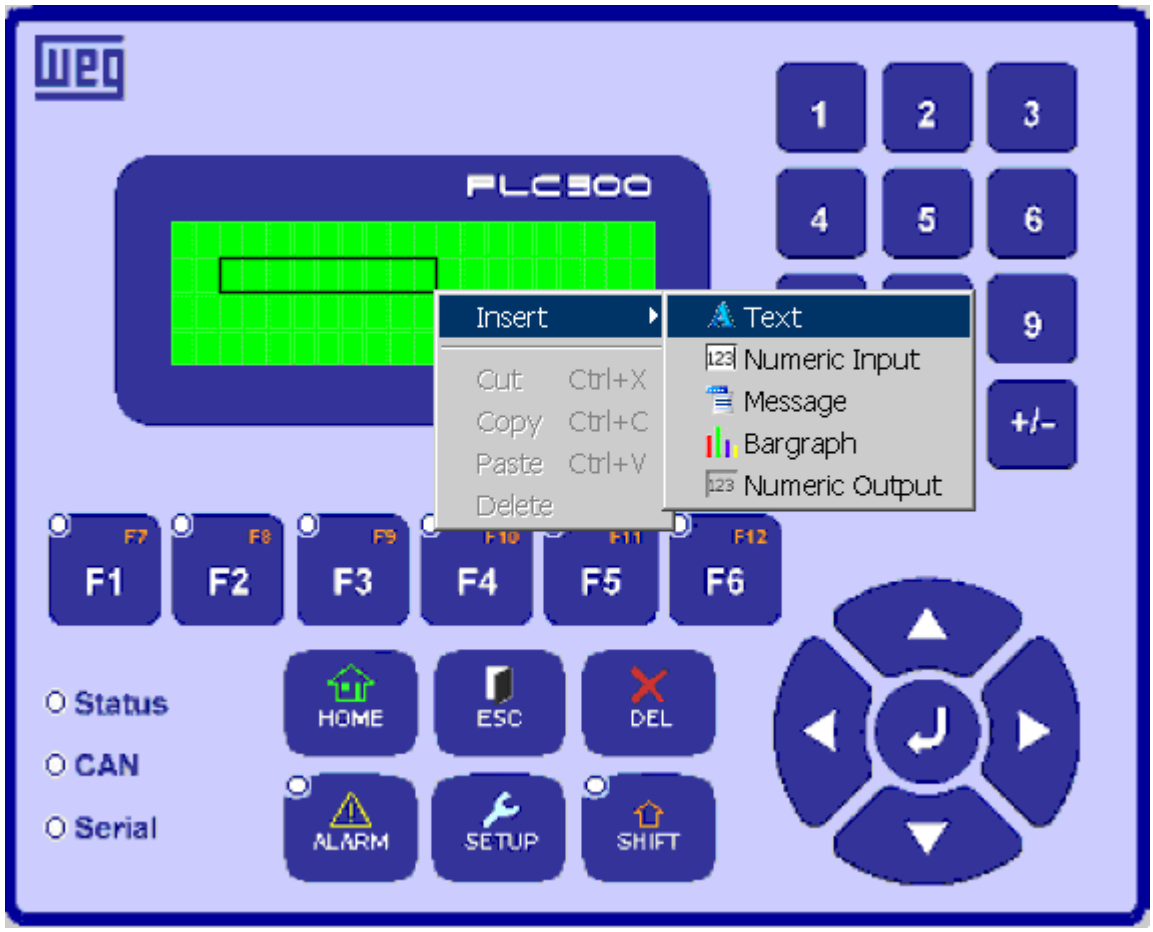
As telas são compostas por componentes do tipo Text , Numeric Input , Numeric Output , Message e Bargraph . Para a programação das telas os componentes são inseridos no display de três maneiras:

Drag'n drop (Arrastar e Soltar)



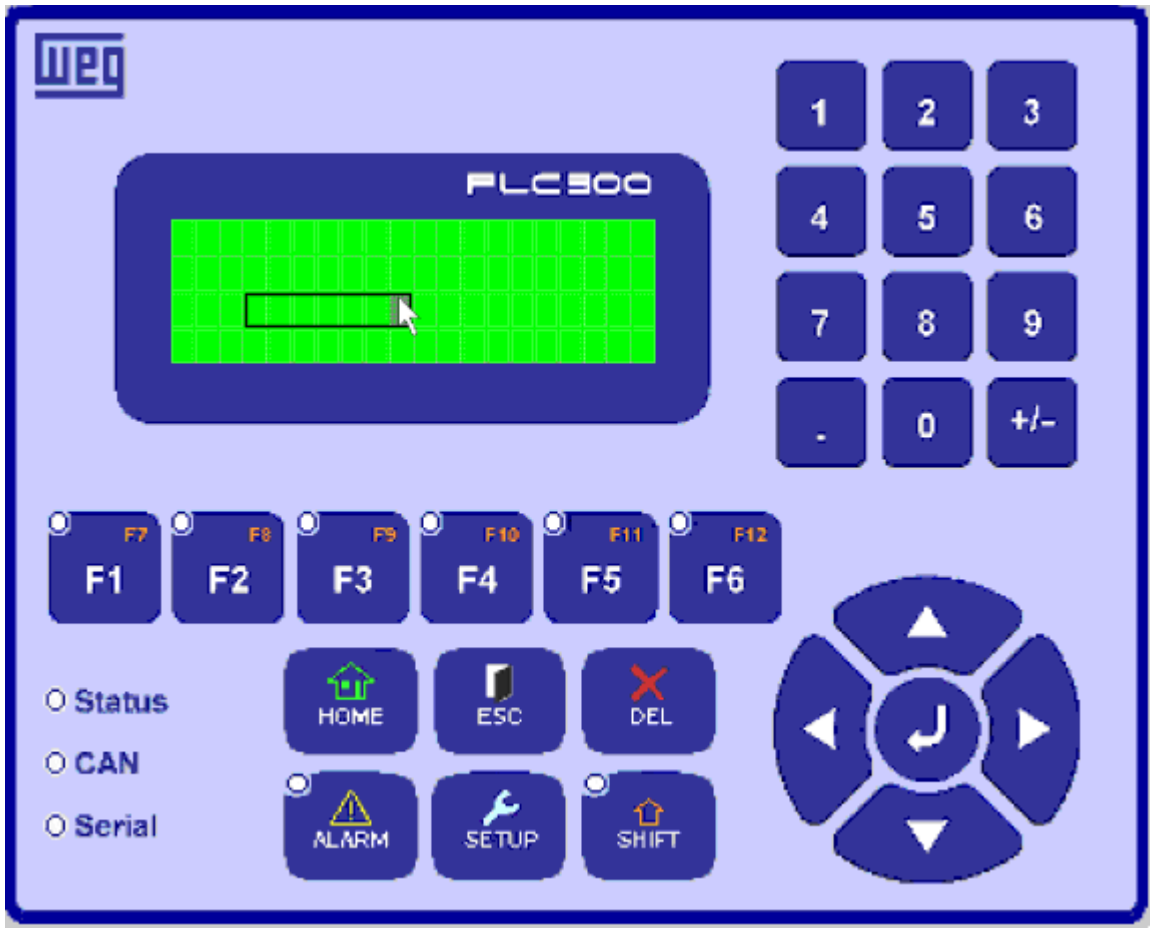
- Selecione o componente desejado na paleta pressionando o botão esquerdo do mouse,
- Mantenha o botão pressionado e arraste até a área do display na qual deseja inserir o componente,
- Certifique-se de que a sombra do componente está na cor cinza (caso contrário o componente não está posicionado corretamente), e
- Solte o botão do mouse.

Teclado



- Com o cursor selecionado dentro da área do display, pressione a tecla Ctrl,
- Seleciona-se o tamanho do componente com as teclas de direção ▲ ► ▼ ◄,
- Pressiona-se Ctrl + Space e aparecerá um pop-up contendo o item inserir, e
- No item inserir, selecione o componente desejado.

Mouse

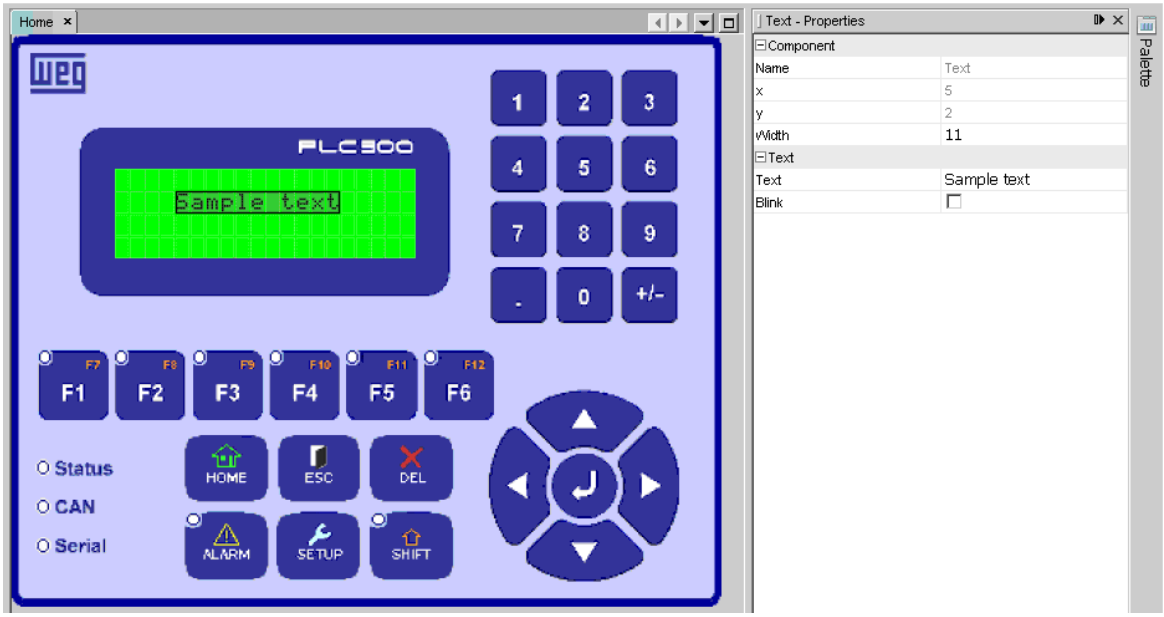


- Selecione uma área no display clicando e arrastando sobre o display,
- Pressione o botão direito do mouse sobre a área selecionada e aparecerá um pop-up contendo o item inserir, e
- No item inserir, selecione o componente desejado.

Configuração dos Componentes

A configuração dos componentes e das propriedades do display é feita através da janela de propriedades. Para tornar as propriedades de um componente ativas, o componente deve estar selecionado na área do display.

7.10 Componente Text

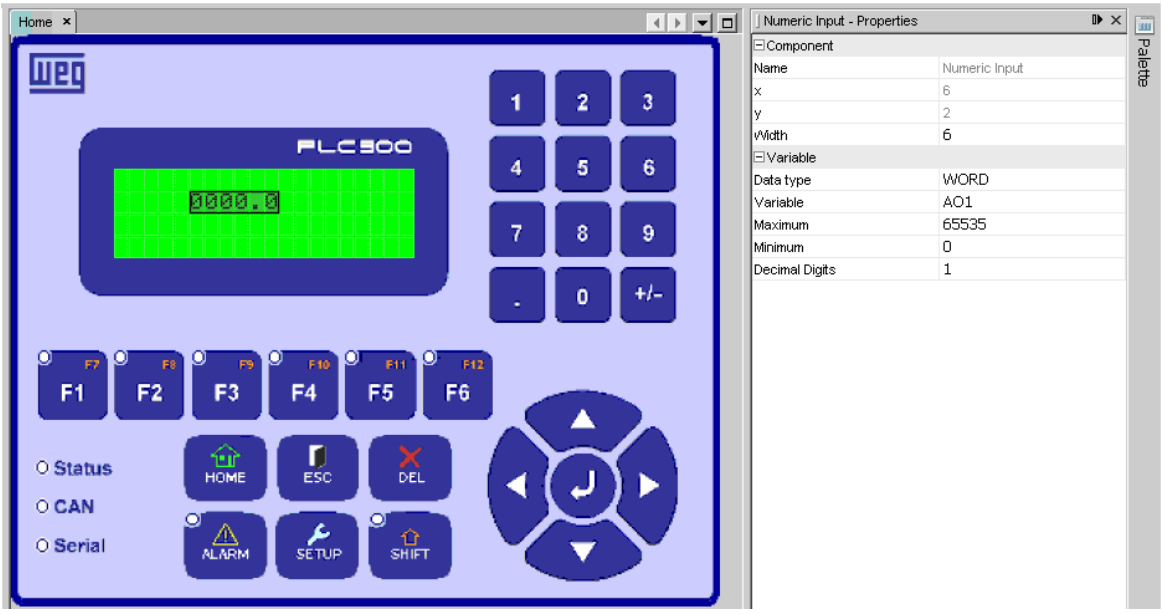


O componente Text é utilizado para inserir textos fixos que na tela. As propriedades deste componente são:

- Componente
  - Nome: Nome do componente (somente leitura),
  - Linha: Número da linha do componente (somente leitura),
  - Coluna: Numero da coluna do componente (somente leitura),
  - Tamanho: Número de células do display ocupadas pelo componente,

- Texto
  - Texto: Texto apresentado na tela,
  - Piscante: Habilita a opção de texto piscante.

7.11 Componente Numeric Input

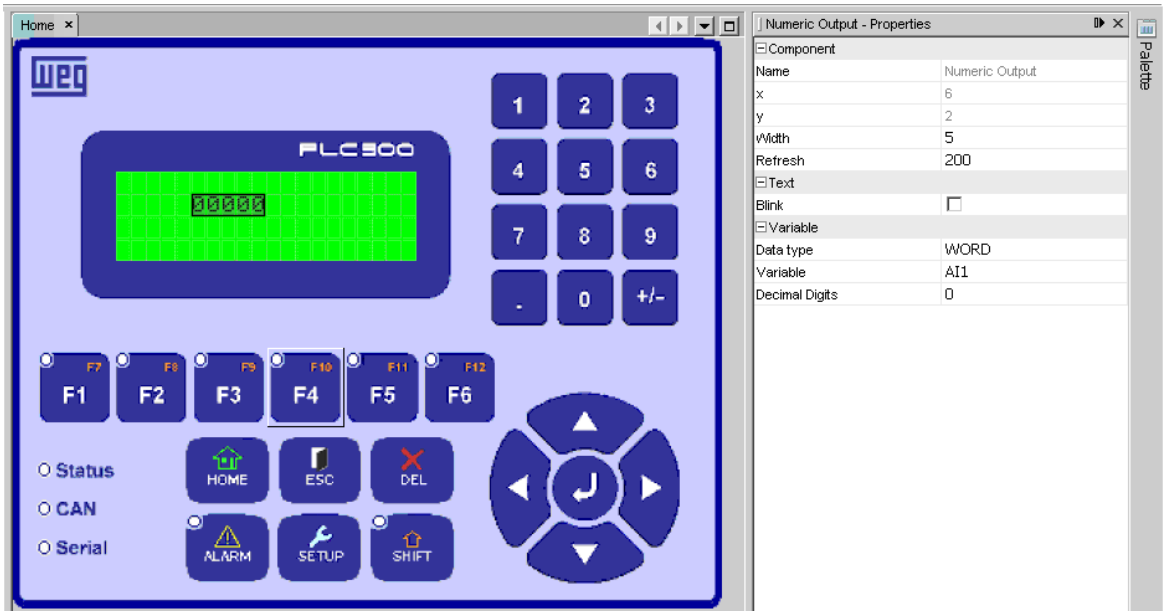


O componente Numeric Input possibilita a entrada de um valor numérico pelo usuário através do PLC 300. O conteúdo digitado é armazenado no marcador configurado na tabela de variáveis do editor ladder. As propriedades deste componente são:

- Componente
  - Nome: Nome do componente (somente leitura),
  - Linha: Número da linha do componente (somente leitura),
  - Coluna: Numero da coluna do componente (somente leitura),
  - Tamanho: Número de células do display ocupadas pelo componente,

- Variável
  - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD), DINT, UDINT(DWORD) e REAL,
  - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima,
  - Máximo: Valor máximo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor máximo, e
  - Mínimo: Valor mínimo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor mínimo.

7.12 Componente Numeric Output

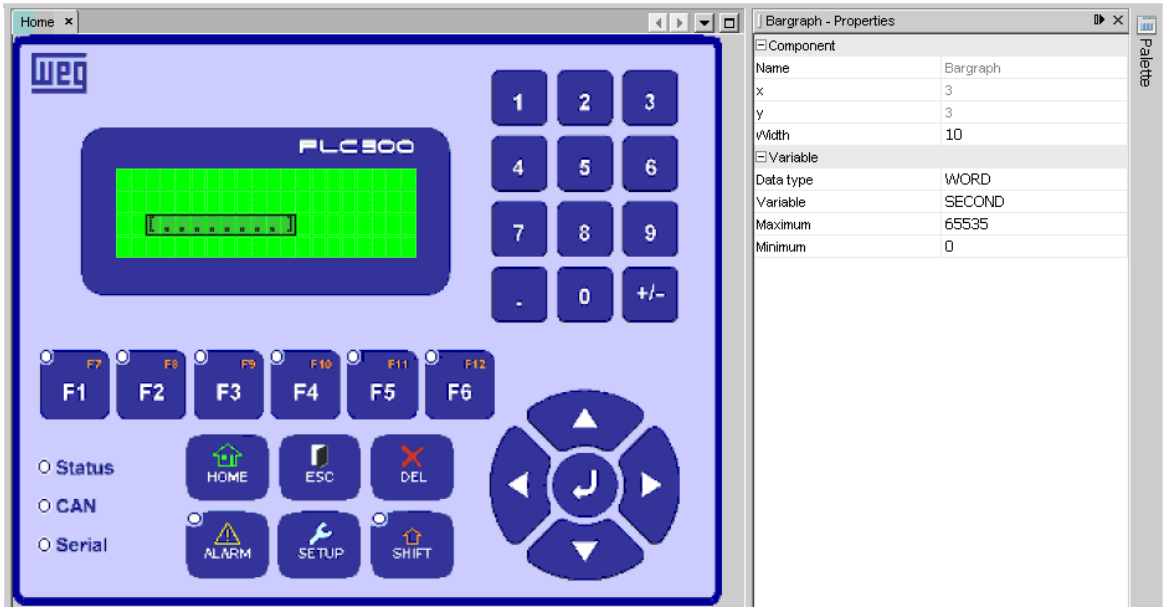


O componente Numeric Output 123 apresenta o valor formatado da variável selecionada no display. As propriedades deste componente são:

- Componente
  - Nome: Nome do componente (somente leitura),
  - Linha: Número da linha do componente (somente leitura),
  - Coluna: Numero da coluna do componente (somente leitura),
  - Tamanho: Número de células do display ocupadas pelo componente,

- Variável
  - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD), DINT, UDINT(DWORD) e REAL,
  - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima, e
  - Dígitos decimais: Formatação do número de casas decimais na apresentação da variável no display.

7.13 Componente Bargraph

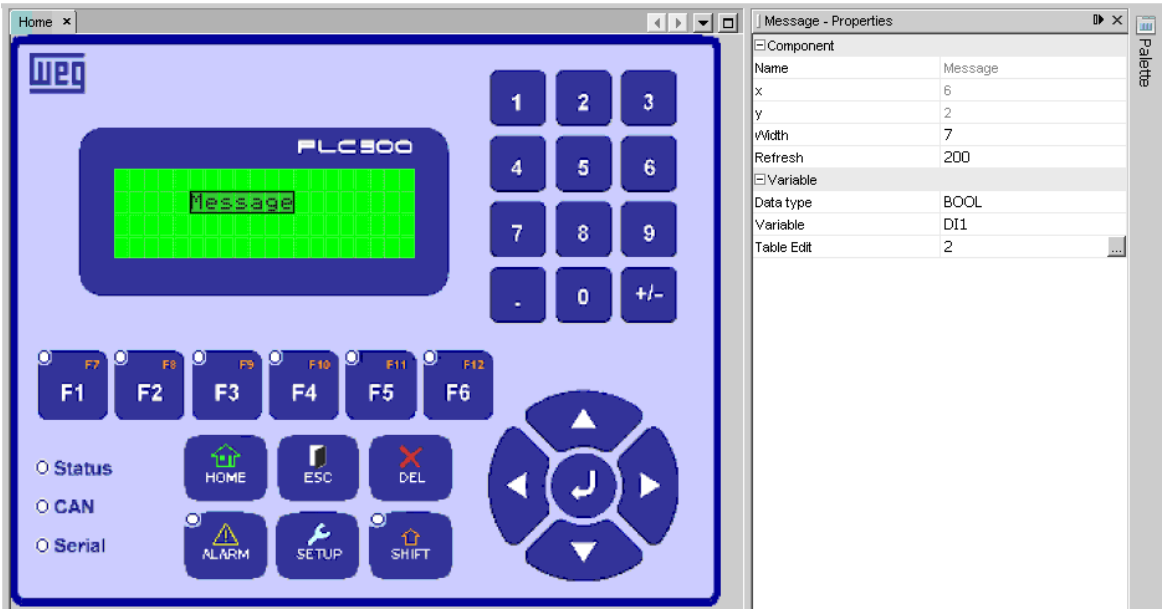


O componente Bargraph mostra uma barra construída com caracteres tipo bloco, proporcional ao valor da variável. As propriedades deste componente são:

- Componente
  - Nome: Nome do componente (somente leitura),
  - Linha: Número da linha do componente (somente leitura),
  - Coluna: Numero da coluna do componente (somente leitura),
  - Tamanho: Número de células do display ocupadas pelo componente,

- Variável
  - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD).
  - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima.
  - Máximo: Valor máximo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor máximo.
  - Mínimo: Valor mínimo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor mínimo.

7.14 Componente Message



O componente Message apresenta textos configuráveis para valores de uma variável. . As propriedades deste componente são:

- Componente
  - Nome: Nome do componente (somente leitura),
  - Linha: Número da linha do componente (somente leitura),
  - Coluna: Numero da coluna do componente (somente leitura),
  - Tamanho: Número de células do display ocupadas pelo componente,

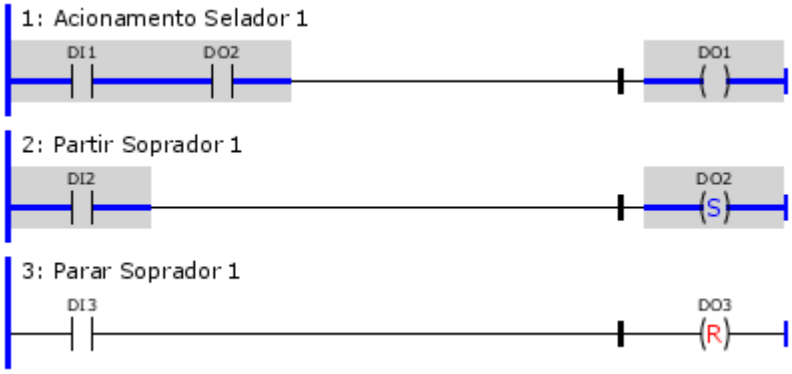
- Variável
  - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD), DINT, UDINT(DWORD) e REAL,
  - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima,
  - Tabela: Valor máximo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor máximo.

8 Monitoração

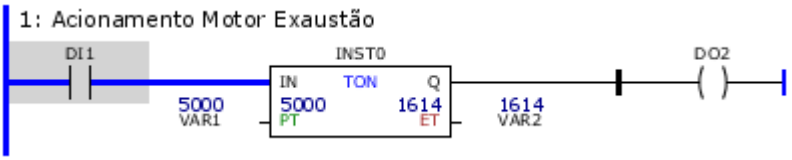
8.1 Monitoração Online Ladder

Após o programa ladder ser compilado e carregado na equipamento é possível monitorar o ladder pressionando o botão de monitoração online . Neste momento o WPS V1.20 tentará estabelecer a comunicação com o equipamento testando a comunicação com o mesmo.

A monitoração online representará graficamente o estado lógico do programa ladder. Um exemplo de monitoração online pode ser vista na figura a seguir.

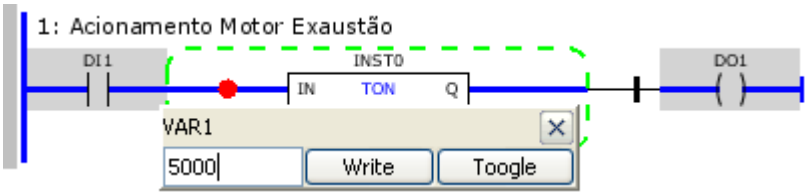


Para blocos funcionais os valores serão apresentados conforme figura a seguir:



Os valores das variáveis são mostrados em cima de suas respectivas variáveis, os valores internos da instância são mostrados em cima do nome do argumento respectivo.

Para escrita de variáveis basta dar um clique duplo sobre a variável que deseja-se escrever e então uma caixa de escrita de valores abrirá conforme figura a seguir.



Nesta caixa deve-se digitar o valor desejado e pressionar o botão "Write" para escrever o valor. O botão "Toogle" serve para alternar o valor escrito de 0 para o valor atual e vice-versa.

9 PLC300

9.1 Descrição

Descrição dos Modelos

O PLC300 está disponível em seis modelos diferentes. Com ou sem HMI; standard ou plus e uma versão com HMI, porém sem membrana, em que o usuário poderá personalizar a aparência do PLC, seguindo especificações presentes no CD do produto.

A identificação do PLC300 é feita por um sufixo de duas ou três letras.

As letras têm o seguinte significado:

- H: HMI
- B: Blind (sem HMI)
- P: Plus
- S: Standard
- C: Custom (sem membrana)

Especificações	PLC300HP	PLC300BP	PLC300HS	PLC300BS	PLC300HPC	PLC300HSC
Cód. WEG	11279219	11279283	11477962	11477953	11490903	11491069
IHM	X	--	X	--	X	X
Membrana	X	--	X	--	-	--
Ethernet	X	X	--	--	X	--
Encoder	X	X	--	--	X	--
Expansão	X	X	X	--	X	--
SD Card	X	X	X	X	X	X
CAN	X	X	X	X	X	X
RS485	X	X	X	X	X	X
RS232	X	X	X	X	X	X
USB	X	X	X	X	X	X
Saída PWM	X	X	X	X	X	X
I/O digital	X	X	X	X	X	X
I/O analógico	X	X	X	X	X	X

Para maiores detalhes sobre o produto consulte o manual do usuário do equipamento PLC300.

9.2 Tipo de Dados

Tipo de Dado	Tamanho	Sinal	Faixa
BOOL	Aloca 1 bit de memória.		0 ou 1

BYTE	Aloca 8 bits de memória (1 byte).		0 a 255
USINT	Aloca 8 bits de memória (1 byte).		0 a 255
SINT	Aloca 8 bits de memória (1 byte).	Sim	-128 a 127
WORD	Aloca 16 bits de memória (2 bytes).		0 a 65535
UINT	Aloca 16 bits de memória (2 bytes).		0 a 65535
INT	Aloca 16 bits de memória (2 bytes).	Sim	-32768 a 32767
DWORD	Aloca 32 bits de memória (4 bytes).		0 a (2^{32} - 1)
UDINT	Aloca 32 bits de memória (4 bytes).		0 a (2^{32} - 1)
DINT	Aloca 32 bits de memória (4 bytes).	Sim	-2^{31} a (2^{31} - 1)
LWORD	Aloca 32 bits de memória (8 bytes).		0 a (2^{64} - 1)
ULINT	Aloca 32 bits de memória (8 bytes).		0 a (2^{64} - 1)
LINT	Aloca 32 bits de memória (8 bytes).	Sim	-2^{63} a (2^{63} - 1)
REAL	Dado em ponto flutuante (com ponto decimal). Aloca 32 bits de memória, conforme padrão IEEE 559.	Sim	$+/-10^{+/-38}$; Precisão = 1 / (2^{23})
LREAL	Dado em ponto flutuante (com ponto decimal). Aloca 64 bits de memória, conforme padrão IEEE 559.	Sim	$+/-10^{+/-308}$; Precisão = 1 / (2^{52})

9.3 Glossário

STACK: Pilha do programa Ladder.

LD (Ladder Diagram): Linguagem gráfica de programação baseada nos diagramas elétricos (contatos e bobinas interconectadas), conforme o fluxo de potência entre os elementos.

SCAN: Ciclo de varredura de um programa.

INSTÂNCIA: Área de memória de um bloco funcional

TAG: Nome da variável

POU: Unidade de Organização de Programas

VAR_IN_OUT: Argumento de entrada e de saída de um bloco de função e/ou bloco funcional. Não permite variáveis CONSTANTES, ENTRADAS FÍSICAS (%I_), ENTRADAS DE REDE (%I_) ou MARCADORES DE SISTEMA DE LEITURA (%S_)

VAR_IN: Argumento de entrada e saída de um bloco de função e/ou bloco funcional.

VAR_OUT: Argumento de saída de um bloco de função e/ou bloco funcional. Não permite variáveis CONSTANTES, ENTRADAS FÍSICAS (%I_), ENTRADAS DE REDE (%I_) ou MARCADORES DE SISTEMA DE LEITURA (%S_)

OVERFLOW: Ocorre quando o resultado de um cálculo matemático, ultrapassa os limites permitidos para o tipo de dado do resultado.

10 Linguagem Ladder

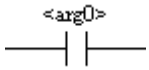
10.1 PLC300

10.1.1 V1.2X

10.1.1.1 Contact

10.1.1.1.1 NOCONTACT

Figura



Configuração do Bloco

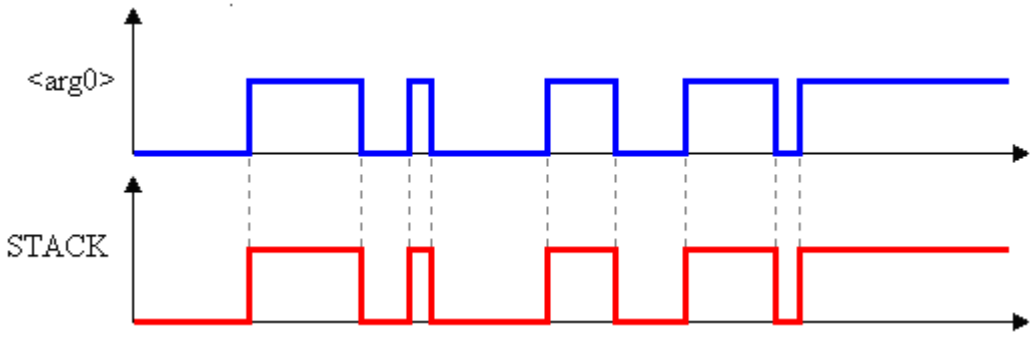
<arg0> - VAR_IN: inserir uma variável (tag).
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega para o STACK o valor de <arg0>.

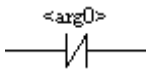
Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

Diagrama de Funcionamento



10.1.1.1.2 NCCONTACT

Figura



Configuração do Bloco

<arg0> - VAR_IN: inserir uma variável (tag).

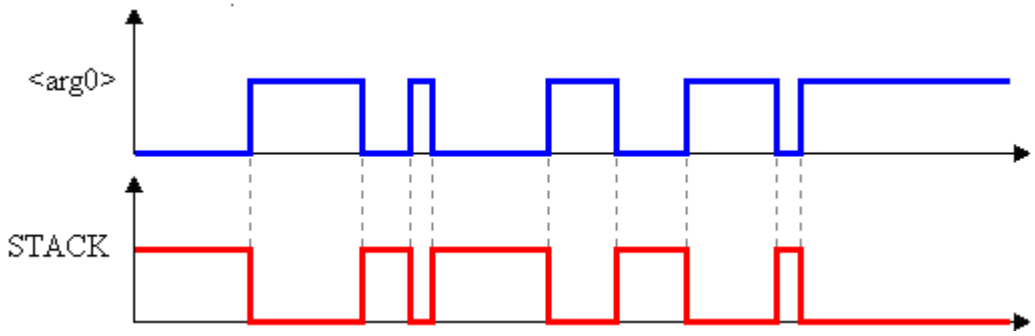
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega para o STACK o valor negado de <arg0>.

Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

Diagrama de Funcionamento



10.1.1.1.3 PTSCONTACT

Figura

Configuração do Bloco

<inst> - inserir uma instância (tag).

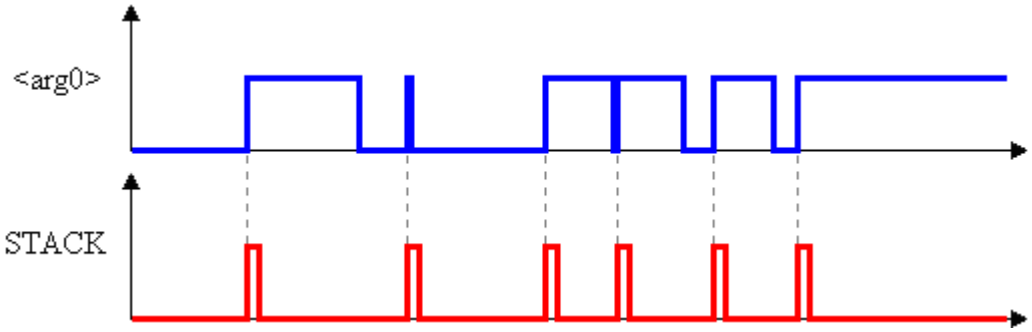
<arg0> - VAR_IN: inserir uma variável (tag).
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega 1 para o STACK por 1 SCAN somente na transição de 0 para 1 de <arg0>, ou seja, na borda de subida do STACK.

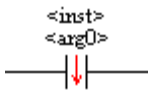
Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

Diagrama de Funcionamento



10.1.1.1.4 NTSCONTACT

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

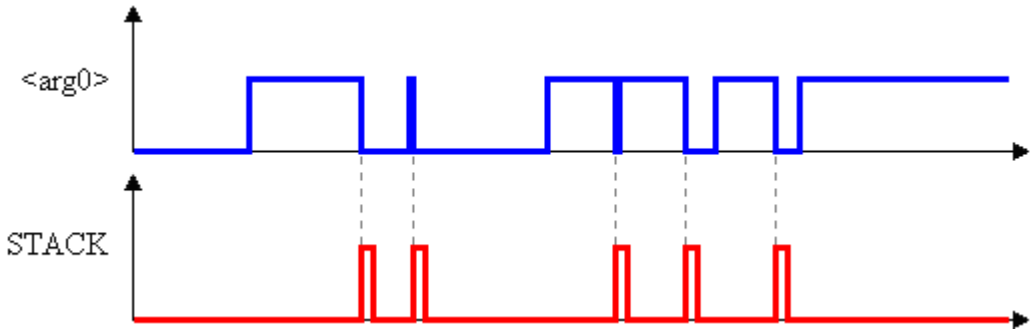
<arg0> - VAR_IN: inserir uma variável (tag).
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega 1 para o STACK por 1 SCAN somente na transição de 1 para 0 de <arg0>, ou seja, na borda de descida do STACK.

Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

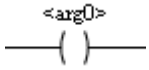
Diagrama de Funcionamento



10.1.1.2 Coil

10.1.1.2.1 DIRECTCOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

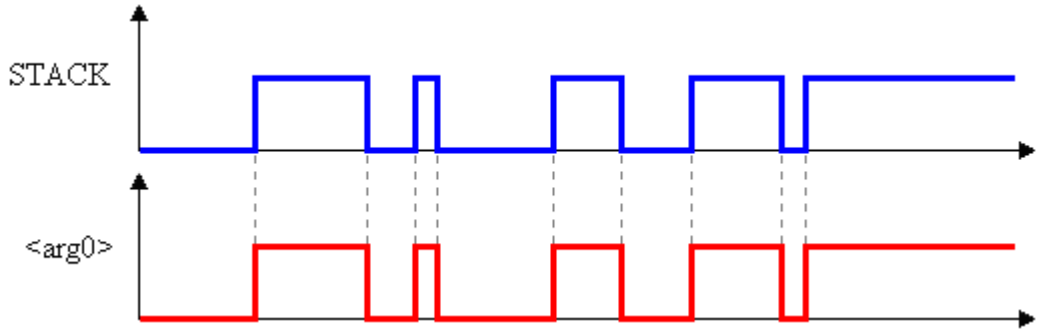
Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

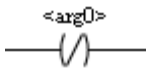
Salva o valor do STACK em <arg0>.

Diagrama de Funcionamento



10.1.1.2.2 NEGATEDCOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

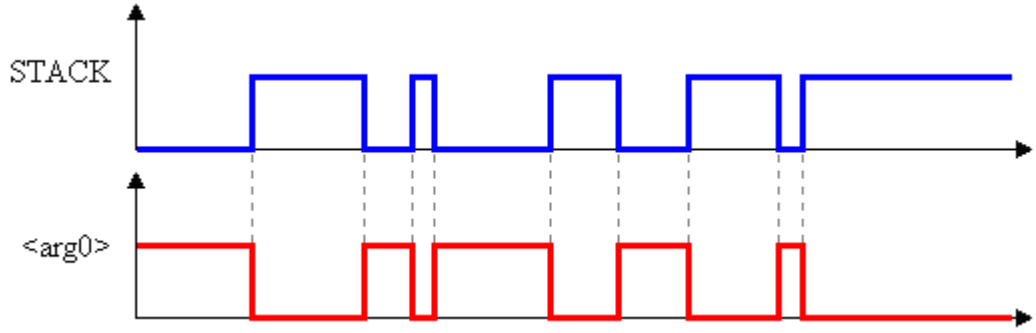
Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

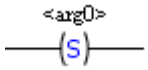
Salva o valor negado do STACK em <arg0>.

Diagrama de Funcionamento



10.1.1.2.3 SETCOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

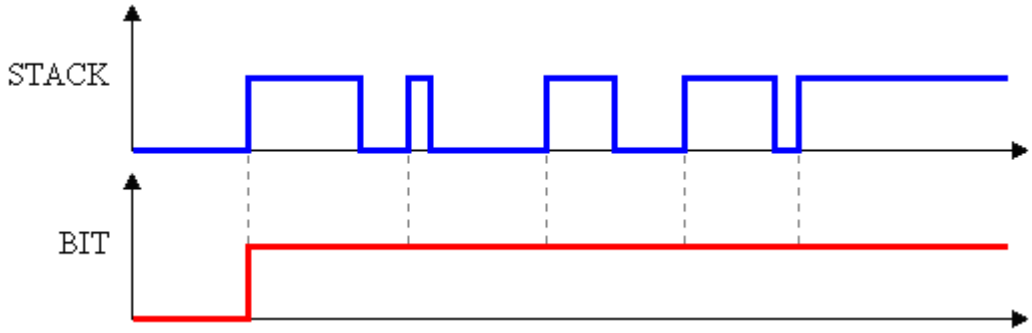
Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

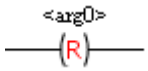
Quando o valor do STACK for 1, <arg0> vai para 1.

Diagrama de Funcionamento



10.1.1.2.4 RESETCOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

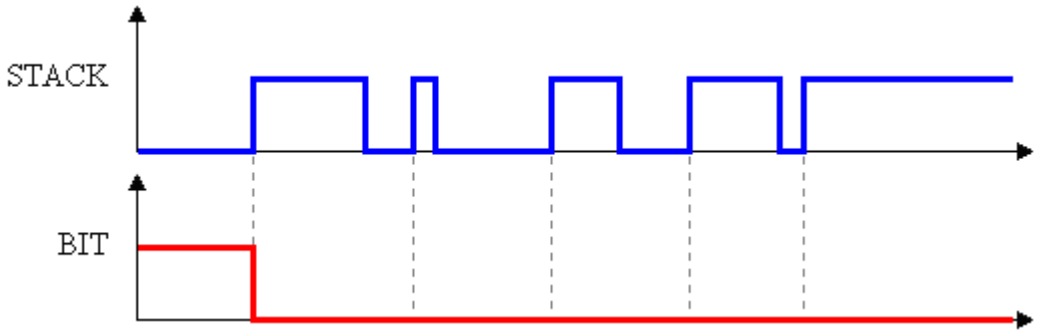
Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

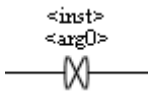
Quando o valor do STACK for 1, <arg0> vai para 0.

Diagrama de Funcionamento



10.1.1.2.5 TOGGLECOIL

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

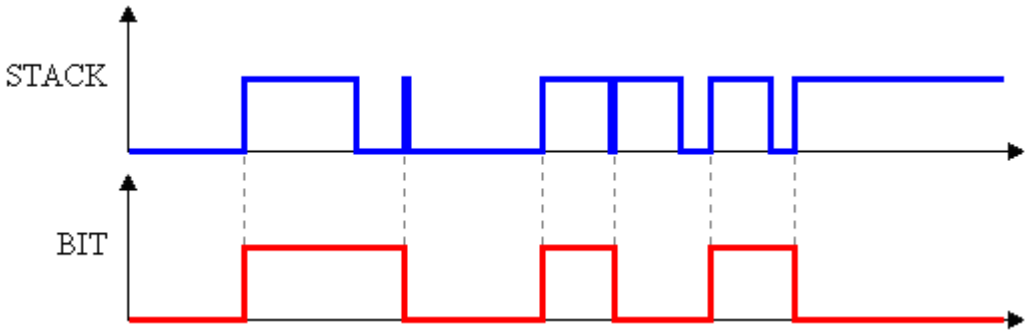
<arg0> - VAR_IN: inserir uma variável (tag).

Tipo de dado: BOOL.

Descrição de Funcionamento

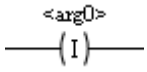
A cada transição de 0 para 1 no STACK, a <arg0> inverte seu valor.

Diagrama de Funcionamento



10.1.1.2.6 IMMEDIATECOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

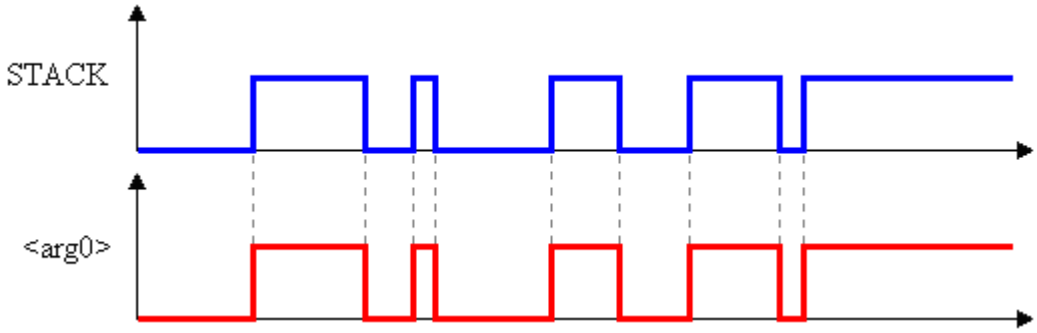
Nota: A variável deve ser uma saída digital do PLC300, excluindo-se as saídas digitais das expansões, isto é, apenas são válidas as DO's de 1 até 9.

Descrição de Funcionamento

Salva o valor do STACK em <arg0>. Diferentemente do DIRECTCOIL, este bloco seta as saídas digitais diretamente, sem aguardar o fim do SCAN do programa. Seu principal uso é nos POU's chamados pelas TASK's single, evento externo, interval e contagem.

A imagem da saída digital também é atualizada imediatamente.

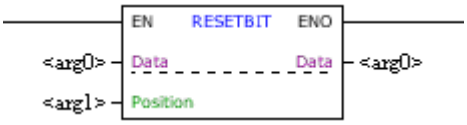
Diagrama de Funcionamento



10.1.1.3 Logic Bit

10.1.1.3.1 RESETBIT

Figura



Configuração do Bloco

<arg0>: "Data" - VAR_IN_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

<arg1>: "Position" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.
Faixa de Valores: 0 a 7 se "Data" for BYTE, USINT ou SINT.
Faixa de Valores: 0 a 15 se "Data" for WORD, UINT ou INT.
Faixa de Valores: 0 a 31 se "Data" for DWORD, UDINT ou DINT.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "Done" normalmente retorna o STACK o valor de "EN". Isto somente não ocorre quando "Position" for configurado de maneira incorreta, mantendo neste caso, "Data" inalterado.

Em execução ("EN"=1), o bit configurado em "Position" da variável em "Data" vai para 0.

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo1 ("EN"=1)

"Position" = 2

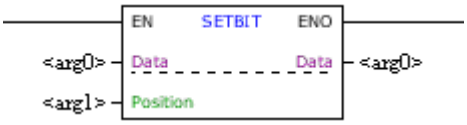
"Data" = 15 = 2#1111 (antes da execução)
"Data" = 11 = 2#1011 (depois da execução)

Exemplo2 ("EN"=1)

"Position" = 0
"Data" = 2 = 2#0010 (antes da execução)
"Data" = 2 = 2#0010 (depois da execução)

10.1.1.3.2 SETBIT

Figura



Configuração do Bloco

<arg0>: "Data" - VAR_IN_OUT: inserir uma variáve (tag)l
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT
Nota: A variável tem que ter permissão de escrita.

<arg1>: "Position" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.
Faixa de Valores: 0 a 7 se "Data" for BYTE, USINT e SINT.
Faixa de Valores: 0 a 15 se "Data" for WORD, UINT e INT.
Faixa de Valores: 0 a 31 se "Data" for DWORD, UDINT, DINT.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "Done" normalmente retorna o STACK o valor de "EN". Isto somente não ocorre quando "Position" for configurado de maneira incorreta, mantendo neste caso, "Data" inalterado.

Em execução ("EN"=1), o bit configurado em "Position" da variável em "Data" vai para 1.

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo1 ("EN"=1)

"Position" = 2
"Data" = 15 = 2#1111 (antes da execução)
"Data" = 15 = 2#1111 (depois da execução)

Exemplo2 ("EN"=1)

"Position" = 0
"Data" = 2 = 2#0010 (antes da execução)
"Data" = 3 = 2#0011 (depois da execução)

10.1.1.3.3 TESTBIT

Figura

Configuração do Bloco

<arg0>: "Data" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT

<arg1>: "Position" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.
Faixa de Valores: 0 a 7 se "Data" for BYTE, USINT e SINT.
Faixa de Valores: 0 a 15 se "Data" for WORD, UINT e INT.
Faixa de Valores: 0 a 31 se "Data" for DWORD, UDINT, DINT.
Nota: "Position" em 0 indica o bit menos significativo da variável.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega para a saída "Q" o valor do bit configurado em "Position" da variável em "Data".

Se "EN" for 0, a saída "Q" vai para 0.

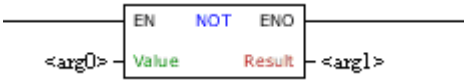
Exemplo1: ("EN"=1)
"Position" = 2
"Data" = 15 = 2#1111 e "Q" = 1

Exemplo2: ("EN"=1)
"Position" = 0
"Data" = 2 = 2#0010 e "Q" = 0

10.1.1.4 Logic Boolean

10.1.1.4.1 NOT

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = NOT "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 12 = 2#1100
"Result" = 3 = 2#0011

Exemplo2 ("EN"=1)
"Value" = 18743 = 2# 0100_1001_0011_0111
"Result" = -56 = 2#1100_1000 (se SINT)
"Result" = 200 = 2#1100_1000 (se USINT)
"Result" = -18744 = 2#1011_0110_1100_1000 (se INT)
"Result" = 46762 = 2#1011_0110_1100_1000 (se UINT)

10.1.1.4.2 AND

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" AND "Value2".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value1" = 12 = 2#1100
"Value2" = 5 = 2#0101
"Result" = 4 = 2#0100

Exemplo2 ("EN"=1)
"Value1" = 18743 = 2# 0100_1001_0011_0111
"Value2" = 43798 = 2#1010_1011_0001_0110
"Result" = 22 = 2#0001_0110 (se SINT)
"Result" = 2326 = 2#0000_1001_0001_0110 (se INT)

10.1.1.4.3 OR

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" OR "Value2".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value1" = 12 = 2#1100.
"Value2" = 5 = 2#0101.
"Result" = 14 = 2#1101.

Exemplo2 ("EN"=1)
"Value1" = 18743 = 2# 0100_1001_0011_0111.
"Value2" = 43798 = 2#1010_1011_0001_0110.
"Result" = 55 = 2#0011_0111 (se SINT).
"Result" = -5321 = 2#1110_1011_0011_0111 (se INT).
"Result" = 60215 = 2#1110_1011_0011_0111 (se UINT).

10.1.1.4.4 XOR

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" XOR "Value2", ou seja, "Result" = ("Value1" AND (NOT "Value2")) OR ((NOT "Value1") AND "Value2").

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value1" = 12 = 2#1100.
"Value2" = 5 = 2#0101.
"Result" = 9 = 2#1001.

Exemplo2 ("EN"=1)
"Value1" = 18743 = 2# 0100_1001_0011_0111.
"Value2" = 43798 = 2#1010_1011_0001_0110.
"Result" = 33 = 2#0010_0001 (se SINT).
"Result" = -7647 = 2#1110_0010_0010_0001 (se INT).
"Result" = 57889 = 2#1110_0010_0010_0001 (se UINT).

10.1.1.4.5 XNOR

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" XNOR "Value2", ou seja, "Result" = ("Value1" AND "Value2") OR ("Value1" AND "Value2").

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value1" = 12 = 2#1100.
"Value2" = 5 = 2#0101.
"Result" = 6 = 2#0110.

Exemplo2 ("EN"=1)
"Value1" = 18743 = 2# 0100_1001_0011_0111.
"Value2" = 43798 = 2#1010_1011_0001_0110.
"Result" = -34 = 2#1101_1110 (se SINT).
"Result" = 222 = 2#1101_1110 (se USINT).
"Result" = 7646 = 2#0001_1101_1101_1110 (se INT).

10.1.1.5 Logic Shift

10.1.1.5.1 ASHL

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: SINT, INT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: SINT, INT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento aritmético binário para a esquerda, conforme "Shift": "Result" = "Value" << "Shift". O sinal de "Value" é

levado em consideração.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 3 = 2#0011
"Shift" = 1
"Result" = 6 = 2#0110

Exemplo2 ("EN"=1)
"Value" = -93 = 2#1010_0011
"Shift" = 4
"Result" = -80 = 2#1011_0000 (se SINT)
"Result" = -1488 = 2#1111_1010_0011_0000 (se INT)

10.1.1.5.2 ASHR

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: SINT, INT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: SINT, INT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento aritmético binário para a direita, conforme "Shift": "Result" = "Value" >> "Shift". O sinal de "Value" é levado em consideração.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 76 = 2#0100_1100
"Shift" = 3
"Result" = 9 = 2#0000_1001 (se SINT)

Exemplo2 ("EN"=1)
"Value" = -71 = 2#1011_1001
"Shift" = 5
"Result" = -3 = 2#1111_1101 (se SINT)

Exemplo3 ("EN"=1)
"Value" = -3011 = 2#1111_0100_0011_1101
"Shift" = 2
"Result" = -113 = 2#1000_1111 (se SINT)
"Result" = -753 = 2#1111_1101_0000_1111 (se INT)

10.1.1.5.3 SHL

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento lógico binário para a esquerda, conforme "Shift": "Result" = "Value" << "Shift".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 163 = 2#1010_0011

"Shift" = 4
"Result" = 48 = 2#0011_0000 (se SINT)
"Result" = 2608 = 2#0000_1010_0011_0000 (se INT)

10.1.1.5.4 SHR

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento lógico binário para a direita, conforme "Shift": "Result" = "Value" >> "Shift".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 163 = 2#1010_0011
"Shift" = 4
"Result" = 10 = 2#0000_1010

Exemplo2 ("EN"=1)
"Value" = 2944 = 2#0000_1011_1000_0000
"Shift" = 3
"Result" = 112 = 2#0111_0000 (se SINT)
"Result" = 368 = 2#0000_0001_0111_0000 (se INT)

10.1.1.6 Logic Rotate

10.1.1.6.1 ROL

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_ OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota1: A variável tem que ter permissão de escrita.
Nota2: O tamanho do tipo de dado deve ser igual ao de "Value".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito uma rotação para a esquerda em "Value" conforme "Shift" e armazenado em "Result". Os bits mais significativos deslocados para fora, são transferidos para a parte menos significativa de "Result".

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 21 = 2#0001_0101.
"Shift" = 5
"Result" = 162 = 2#1010_0010

10.1.1.6.2 ROR

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_ OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota1: A variável tem que ter permissão de escrita.
Nota2: O tamanho do tipo de dado deve ser igual ao de "Value".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito uma rotação para a direita em "Value" conforme "Shift" e armazenado em "Result". Os bits menos significativos deslocados para fora, são transferidos para a parte mais significativa de "Result".

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 21 = 2#0001_0101.
"Shift" = 3.
"Result" = 162 = 2#1010_0010.

10.1.1.7 Math Basic

10.1.1.7.1 ABS

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = |"Value"|. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

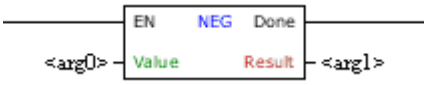
Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = -5.
"Result" = 5 e "Done" = 1.

Exemplo2 ("EN"=1)
"Value" = -200.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 200 (se USINT) e "Done"=1.

10.1.1.7.2 NEG

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: SINT, INT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = - "Value". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = -5
"Result" = 5 e "Done"=1.

Exemplo2 ("EN"=1)
"Value" = 10
"Result" = -10 e "Done"=1.

Exemplo3 ("EN"=1)
"Value" = -200
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 200 (se USINT) e "Done"=1.

10.1.1.7.3 ADD

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1, ou seja, o topo do STACK for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" + "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Em caso contrário ("EN"=0), o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value1" = 100.
"Value2" = 150.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 250 (se USINT) e "Done"=1.

Exemplo2 ("EN"=1)
"Value1" = -100.
"Value2" = -150.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se UINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = -250 (se INT) e "Done"=1.

10.1.1.7.4 SUB

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" - "Value2". Se não ocorrer

erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value1" = 400.
"Value2" = 210.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 190 (se USINT) e "Done"=1.

Exemplo2 ("EN"=1)
"Value1" = 500.
"Value2" = 730.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = -230 (se INT) e "Done"=1.

10.1.1.7.5 MUL

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" * "Value2". Se não ocorrer

erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)
"Value1" = 143.
"Value2" = 308.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se INT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 44044 (se UINT) e "Done"=1.

10.1.1.7.6 DIV

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" / "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)
"Value1" = 751.
"Value2" = 3.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 250 (se USINT) e "Done"=1.
"Result" = 250.333... (se REAL) e "Done"=1.

10.1.1.7.7 MOD

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" % "Value2", ou seja, é calculado o resto entre a divisão de "Value1" com "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

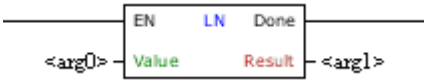
Exemplo1 ("EN"=1)
"Value1" = 7.
"Value2" = 3.
"Result" = 1 e "Done"=1.

Exemplo2 ("EN"=1)
"Value1" = 1000.
"Value2" = 400.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 200 (se USINT) e "Done"=1.

10.1.1.8 Math Extended

10.1.1.8.1 LN

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Faixa de valores: somente valores reais positivos.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = ln("Value"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.
Nota: ln é o log na base = 2.71828182845904523536028 74.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

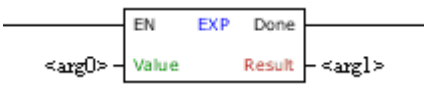
Exemplo1 ("EN"=1)
"Value" = 100.
"Result" = 4.605 e "Done" = 1.

Exemplo2 ("EN"=1)
"Value" = 0.025.
"Result" = -3.689 e "Done"=1.

Exemplo3 ("EN"=1)
"Value" = 0.0 (valor fora da faixa).
"Result" = inalterado e "Done"=0.

10.1.1.8.2 EXP

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = e"Value". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.
Nota: e = 2.71828182845904523536028747.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = 5.
"Result" = 148.413 e "Done" = 1.

Exemplo2 ("EN"=1)
"Value" = -2.
"Result" = 0.135 e "Done"=1.

10.1.1.8.3 LOG10

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Faixa de valores: somente valores reais positivos.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = log10("Value"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = 110.53
"Result" = 2.043 e "Done" = 1.

Exemplo2 ("EN"=1)
"Value" = 0.147.
"Result" = -0.883 e "Done"=1.

Exemplo3 ("EN"=1)
"Value" = -1.3 (valor fora da faixa).
"Result" = inalterado e "Done"=0.

10.1.1.8.4 ALOG10

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = 10"Value". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = 2.5.
"Result" = 316.228 e "Done" = 1.

Exemplo2 ("EN"=1)

"Value" = -1.333.
"Result" = 0.046 e "Done"=1.

10.1.1.8.5 POW

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Power" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value" "Power". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = 2.
"Power" = 3.
"Result" = 8 e "Done"=1.

Exemplo2 ("EN"=1)
"Value" = -49.
"Power" = 0.5.
"Result" = inalterado e "Done"=0 (resulta num valor imaginário).

10.1.1.8.6 SQRT

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Faixa de valores: somente valores reais positivos.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = RaizQuadrada("Value"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

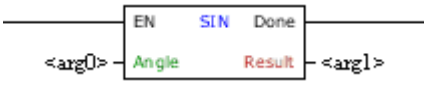
Exemplo1 ("EN"=1)
"Value" = 81.
"Result" = 9. e "Done" = 1.

Exemplo2 ("EN"=1)
"Value" = -4.
"Result" = inalterado e "Done"=0 (resulta num valor imaginário).

10.1.1.9 Math Trigonometry

10.1.1.9.1 SIN

Figura



Configuração do Bloco

<arg0>: "Angle" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = sin("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)
"Angle" = 1.047 (equivalente a 60º).
"Result" = 0.866 e "Done"=1.

10.1.1.9.2 COS

Figura

Configuração do Bloco

<arg0>: "Angle" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

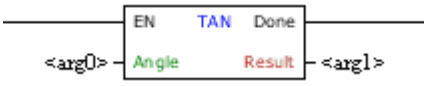
Em execução ("EN"=1), é calculado: "Result" = cos("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)
"Angle" = 1.047 (equivalente a 60º).
"Result" = 0.5 e "Done"=1.

10.1.1.9.3 TAN

Figura



Configuração do Bloco

<arg0>: "Angle" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

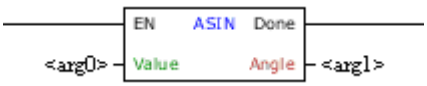
Em execução ("EN"=1), é calculado: "Result" = tan("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)
"Angle" = 1.047 (equivalente a 60º).
"Result" = 1.732 e "Done"=1.

10.1.1.9.4 ASIN

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Faixa de Valores: -1.0 a 1.0.

<arg1>: "Angle" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = arc-sin("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = 0.866.
"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

Exemplo2 ("EN"=1)
"Value" = 1.5 (valor fora da faixa permitida).
"Angle" = inalterado e "Done"=0.

10.1.1.9.5 ACOS

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Faixa de Valores: -1.0 a 1.0.

<arg1>: "Angle" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = arc-cos("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

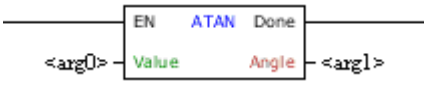
Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)
"Value" = 0.5.
"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

Exemplo2 ("EN"=1)
"Value" = 1.5 (valor fora da faixa permitida).
"Angle" = inalterado e "Done"=0.

10.1.1.9.6 ATAN

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Angle" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = arc-tan("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)
"Value" = 1.732.
"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

10.1.1.9.7 ATAN2

Figura

Configuração do Bloco

<arg0>: "X" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg0>: "Y" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg2>: "Angle" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = arc-tan("Y" / "X"). Diferentemente do bloco ATAN, este bloco leva em consideração o quadrante para calcular o valor do ângulo. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)
"X" = 1.
"Y" = 1.732.
"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

10.1.1.10 Math Util

10.1.1.10.1 MAX

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1.

Em execução (EN=1), é calculado: Se "Value1" > "Value2", "Result" = "Value1", senão "Result"="Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Em caso contrário (EN=0), o valor em "Result" permanece inalterado.

Exemplo1 (considerando EN=1)
"Value1" = 100.
"Value2" = 150.
"Result" = 150.

Exemplo2 (considerando EN=1)
"Value1" = -1000.
"Value2" = -1500.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se UINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = -1000 (se INT) e "Done"=1.

10.1.1.10.2 MIN

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou

REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1.

Em execução (EN=1), é calculado: Se "Value1" < "Value2", "Result" = "Value1", senão "Result"="Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Em caso contrário (EN=0), o valor em "Result" permanece inalterado.

Exemplo1 (considerando EN=1)
"Value1" = 100.
"Value2" = 150.
"Result" = 100.

Exemplo2 (considerando EN=1)
"Value1" = -1000.
"Value2" = -1500.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se UINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = -1500 (se INT) e "Done"=1.

10.1.1.10.3 SAT

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou

REAL.

<arg1>: "Minimum" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Maximum" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg3>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1.

Em execução ("EN"=1), a saída "Q" vai para 1 se houver saturação, ou seja: Se "Value" > "Maximum", "Result" = "Maximum" e "Q"=1; Se Value < "Minimum", Result = "Minimum" e "Q"=1; Senão "Result"="Value" e "Q"=0. Se ocorrer erros, "Q" indica 0 e o "Result" permanece inalterado.

Em caso contrário ("EN"=0), o valor em "Result" permanece inalterado.

Exemplo1 (considerando "EN"=1)
"Value" = 10.
"Minimum" = 5.
"Maximum" = 15.
"Result" = 10 e "Q"=0.

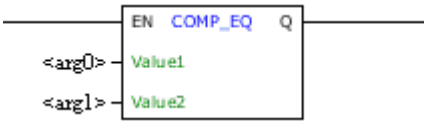
Exemplo2 (considerando "EN"=1)
"Value" = 30.
"Minimum" = 5.
"Maximum" = 15.
"Result" = 15 e "Q"=1.

Exemplo3 (considerando "EN"=1)
"Value" = 2.
"Minimum" = 5.
"Maximum" = 15.
"Result" = 5 e "Q"=1.

10.1.1.11 Compare

10.1.1.11.1 COMPEQ

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"="Value2. Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

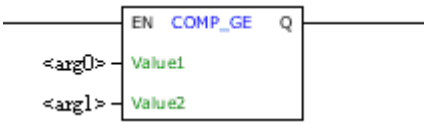
Exemplo1: ("EN"=1)
"Value1" = 2.3
"Value2" = 2.3
"Q" = 1

Exemplo2: ("EN"=1)
"Value1" = 4
"Value2" = 26
"Q" = 0

Exemplo3: ("EN"=1)
"Value1" = 4158
"Value2" = 36.5
"Q" = 0

10.1.1.11.2 COMPGE

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"="Value2. Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)
"Value1" = 2.3
"Value2" = 2.3
"Q" = 1

Exemplo2: ("EN"=1)
"Value1" = 4
"Value2" = 26
"Q" = 0

Exemplo3: ("EN"=1)
"Value1" = 4158
"Value2" = 36.5
"Q" = 1

10.1.1.11.3 COMPGT

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1">"Value2. Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)
"Value1" = 2.3
"Value2" = 2.3
"Q" = 0

Exemplo2: ("EN"=1)
"Value1" = 4
"Value2" = 26
"Q" = 0

Exemplo3: ("EN"=1)
"Value1" = 4158
"Value2" = 36.5
"Q" = 1

10.1.1.11.4 COMPLE

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"="Value2. Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)
"Value1" = 2.3
"Value2" = 2.3
"Q" = 1

Exemplo2: ("EN"=1)
"Value1" = 4
"Value2" = 26
"Q" = 1

Exemplo3: ("EN"=1)
"Value1" = 4158
"Value2" = 36.5
"Q" = 0

10.1.1.11.5 COMPLT

Figura

Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"<"Value2. Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

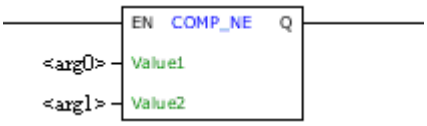
Exemplo1: ("EN"=1)
"Value1" = 2.3
"Value2" = 2.3
"Q" = 0

Exemplo2: ("EN"=1)
"Value1" = 4
"Value2" = 26
"Q" = 1

Exemplo3: ("EN"=1)
"Value1" = 4158
"Value2" = 36.5
"Q" = 0

10.1.1.11.6 COMPNE

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"?"Value2. Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)
"Value1" = 2.3
"Value2" = 2.3
"Q" = 0

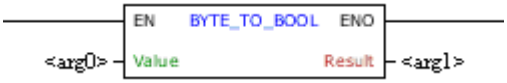
Exemplo2: ("EN"=1)
"Value1" = 4
"Value2" = 26
"Q" = 1

Exemplo3: ("EN"=1)
"Value1" = 4158
"Value2" = 36.5
"Q" = 1

10.1.1.12 Conversion To Bool

10.1.1.12.1 BYTE_TO_BOOL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BYTE para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1. Senão, "Result"=0.

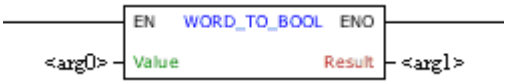
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)
"Value" = 6.
"Result" = 1.

10.1.1.12.2 WORD_TO_BOOL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de WORD para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1. Senão, "Result"=0.

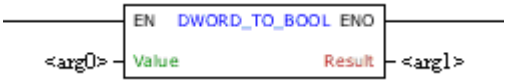
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)
"Value" = 6.
"Result" = 1.

10.1.1.12.3 DWORD_TO_BOOL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1. Senão, "Result"=0.

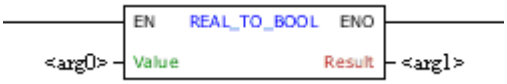
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)
"Value" = 6.
"Result" = 1.

10.1.1.12.4 REAL_TO_BOOL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1.
Senão, "Result"=0.

Se "EN" for 0, o valor em "Result" permanece inalterado.

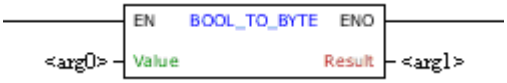
Exemplo1 ("EN"=1)
"Value" = 0.0.
"Result" = 0.

Exemplo2 ("EN"=1)
"Value" = -0.26.
"Result" = 1.

10.1.1.13 Conversion To Byte

10.1.1.13.1 BOOL_TO_BYTE

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para BYTE.

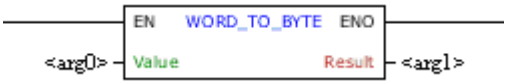
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)
"Value" = 1.
"Result" = 1.

10.1.1.13.2 WORD_TO_BYTE

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de WORD para BYTE.

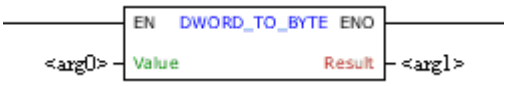
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 145.
"Result" = 145 e "Done"=1.

Exemplo2 ("EN"=1)
"Value" = 1452 = 16#5AC.
"Result" = -84 = 16#AC (se INT).
"Result" = 172 = 16#AC (se UINT).

10.1.1.13.3 DWORD_TO_BYTE

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para BYTE.

Se "EN" for 0, o valor em "Result" permanece inalterado.

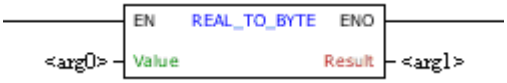
Exemplo1 ("EN"=1)

"Value" = 145.
"Result" = 145 e "Done"=1.

Exemplo2 ("EN"=1)
"Value" = 1452 = 16#5AC.
"Result" = -84 = 16#AC (se INT).
"Result" = 172 = 16#AC (se UINT).

10.1.1.13.4 REAL_TO_BYTE

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para BYTE.

Se "EN" for 0, o valor em "Result" permanece inalterado.

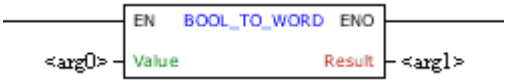
Exemplo1 ("EN"=1)
"Value" = -45.6.
"Result" = -45 = 16#D3 (se SINT).
"Result" = 211 = 16#D3 (se USINT).

Exemplo2 ("EN"=1)
"Value" = 1452.25.
"Result" = -84 = 16#AC (se SINT).
"Result" = 172 = 16#AC (se USINT).

10.1.1.14 Conversion To Word

10.1.1.14.1 BOOL_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para WORD.

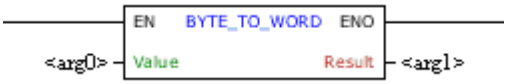
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)
"Value" = 1.
"Result" = 1.

10.1.1.14.2 BYTE_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

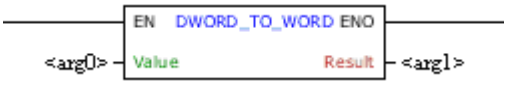
Em execução ("EN"=1), é executada a conversão de BYTE para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 145.
"Result" = 145.

10.1.1.14.3 DWORD_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

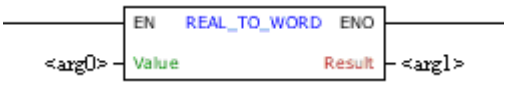
Exemplo1 ("EN"=1)
"Value" = 145
"Result" = 145

Exemplo2 ("EN"=1)
"Value" = 104758

"Result" = -26314= 16# 9936 (se INT)
"Result" = 39222 = 16# 9936 (se UINT)

10.1.1.14.4 REAL_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

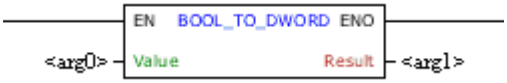
Exemplo1 ("EN"=1)
"Value" = -45.74.
"Result" = -45 = 16#FFD3 (se INT).
"Result" = 65491= 16#FFD3 (se UINT).

Exemplo2 ("EN"=1)
"Value" = 104386.234.
"Result" = -26686= 16#97C2 (se INT).
"Result" = 38850= 16#97C2 (se UINT).

10.1.1.15 Conversion To Dword

10.1.1.15.1 BOOL_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para DWORD.

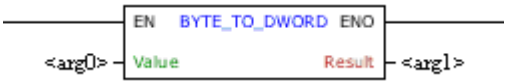
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)
"Value" = 1.
"Result" = 1.

10.1.1.15.2 BYTE_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

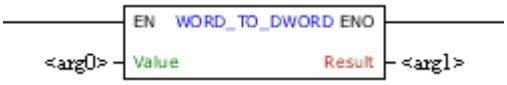
Em execução ("EN"=1), é executada a conversão de BYTE para DWORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 145.
"Result" = 145.

10.1.1.15.3 WORD_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

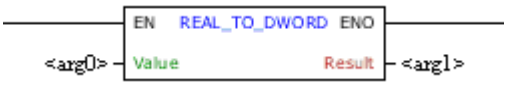
Em execução ("EN"=1), é executada a conversão de WORD para DWORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 145.
"Result" = 145.

10.1.1.15.4 REAL_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para DWORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

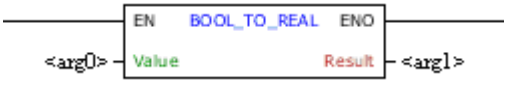
Exemplo1 ("EN"=1)
"Value" = -45.75
"Result" = -45 = 16#FFFF_FFD3 (se DINT).
"Result" = 4294967251 = 16#FFFF_FFD3 (se UDINT).

Exemplo2 ("EN"=1)
"Value" = 11275503078.234.
"Result" = -1609398810 = 16#A012_85E6 (se DINT).
"Result" = 2685568486 = 16#A012_85E6 (se UDINT).

10.1.1.16 Conversion To Real

10.1.1.16.1 BOOL_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para REAL, ou seja, para qualquer "Value" diferente de 0, "Result"=1.
Senão, "Result"=0.

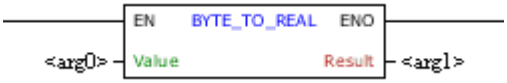
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Value" = 0.
"Result" = 0.0.

Exemplo2 ("EN"=1)
"Value" = 1.
"Result" = 1.0

10.1.1.16.2 BYTE_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

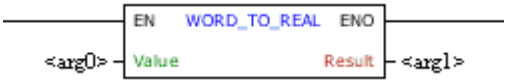
Em execução ("EN"=1), é executada a conversão de BYTE para REAL.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = -45.
"Result" = -45.0.

10.1.1.16.3 WORD_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

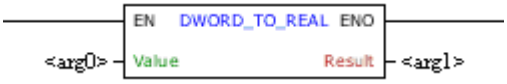
Em execução ("EN"=1), é executada a conversão de WORD para REAL.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = -45.
"Result" = -45.0.

10.1.1.16.4 DWORD_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para REAL.

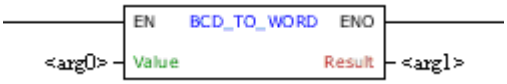
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = -45
"Result" = -45.0.

10.1.1.17 Conversion To BCD

10.1.1.17.1 BCD_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

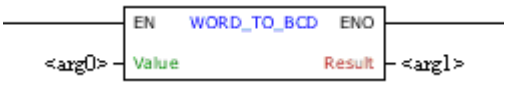
Em execução ("EN"=1), é executada a conversão de WORD para BCD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 123 = 16#7B.
"Result" = 291 = 16#123.

10.1.1.17.2 WORD_TO_BCD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BCD para WORD.

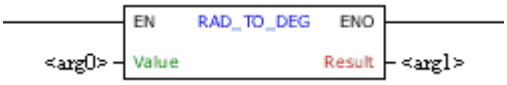
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 291 = 16#123.
"Result "= 123 = 16#7B.

10.1.1.18 Conversion Rad-Deg

10.1.1.18.1 RAD_TO_DEG

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.

Unidade: graus
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

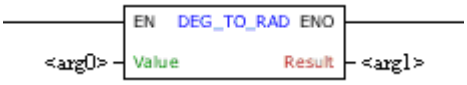
Em execução ("EN"=1), é executada a conversão de radianos para graus.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 1.047.
"Result" = 60.

10.1.1.18.2 DEG_TO_RAD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: graus

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de graus para radianos.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Value" = 60.
"Result" = 1.047.

10.1.1.19 Data Transfer

10.1.1.19.1 STORE

Figura

Configuração do Bloco

<arg0>: "SRC" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Descrição: dado fonte.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT se "SRC" for BYTE, USINT ou SINT.
Tipo de dado: WORD, UINT ou INT se "SRC" for WORD, UINT ou INT.
Tipo de dado: DWORD, UDINT, DINT ou REAL se "SRC" for DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.
Descrição: dado destino.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o conteúdo de "SRC" é armazenado em "DST". O tamanho do tipo de dado de "DST" deve ser o mesmo que o de "SRC".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"SRC" = 145 (SINT).
"DST" = 145 (USINT).

Exemplo2 ("EN"=1)
"SRC" = 4.85 (REAL).
"DST" = 859020096 = 16#3333_9B40 (DINT).

10.1.1.19.2 ILOADBOOL

Figura

Configuração do Bloco

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
Opções: GLOBAL, GLOBAL_RETAIN, SYS_STATUS, SYS_COMMAND, HW_INPUT, HW_OUTPUT, NET_INPUT ou NET_OUTPUT.
Descrição: grupo em que variável global está armazenada.

<arg1>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Descrição: endereço da variável global, conforme o seu grupo.

<arg2>: "Bit" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Descrição: posição do bit a ser verificado da variável global.

<arg3>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.
Descrição: variável com o conteúdo lido.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

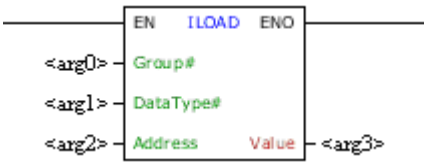
Em execução ("EN"=1), o conteúdo da variável global localizada no "Bit" de "Address" do "Group# é transferido para "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Group#"=HW_OUTPUT
"Address"=0
"Bit" = 4.
"Value"=indicará o conteúdo da saída digital 5 (DO5).

10.1.1.19.3 ILOAD

Figura



Configuração do Bloco

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
Opções: GLOBAL, GLOBAL_RETAIN, SYS_STATUS, SYS_COMMAND, HW_INPUT, HW_OUTPUT, NET_INPUT ou NET_OUTPUT.
Descrição: grupo em que variável global está armazenada.

<arg1>: "DataType#" - VAR_IN: selecionar uma das seguintes opções.
Opções: BYTE, USINT, SINT, WORD UINT, INT, DWORD UDINT, DINT ou REAL.
Descrição: tipo de dado da variável global.

<arg2>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Descrição: endereço da variável global, conforme o seu grupo.

<arg3>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: conforme selecionado em "DataType#".
Nota: A variável tem que ter permissão de escrita.
Descrição: variável com o conteúdo lido.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o conteúdo da variável global localizada em "Address" do "Group#, que é do tipo "DataType#", é transferido para "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Group#"=HW_INPUT
"DataType#"=WORD
"Address"=6
"Value"=indicará o conteúdo da entrada analógica 1 (AI1).

10.1.1.19.4 ISTOREBOOL

Figura

Configuração do Bloco

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
Opções: GLOBAL, GLOBAL_RETAIN, SYS_COMMAND, HW_OUTPUT ou NET_OUTPUT.
Descrição: grupo em que variável global está armazenada.

<arg1>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Descrição: endereço da variável global, conforme o seu grupo.

<arg2>: "Bit" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Descrição: posição do bit a ser verificado da variável global.

<arg3>: "Value" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipo de dado: BOOL.
Descrição: valor a ser escrito.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

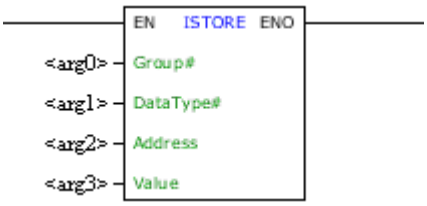
Em execução ("EN"=1), "Value" é transferido para a variável global localizada no "Bit" de "Address" do "Group#.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Group#"=HW_OUTPUT
"Address"=0
"Bit" = 4.
"Value"=1. Seta a saída digital 5 (DO5).

10.1.1.19.5 ISTORE

Figura



Configuração do Bloco

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
Opções: GLOBAL, GLOBAL_RETAIN, SYS_COMMAND, HW_OUTPUT ou NET_OUTPUT.
Descrição: grupo em que variável global está armazenada.

<arg1>: "DataType#" - VAR_IN: selecionar uma das seguintes opções.
Opções: BYTE, USINT, SINT, WORD UINT, INT, DWORD UDINT, DINT ou REAL.
Descrição: tipo de dado da variável global.

<arg2>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Descrição: endereço da variável global, conforme o seu grupo.

<arg3>: "Value" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipo de dado: conforme selecionado em "DataType#".
Descrição: valor a ser escrito.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

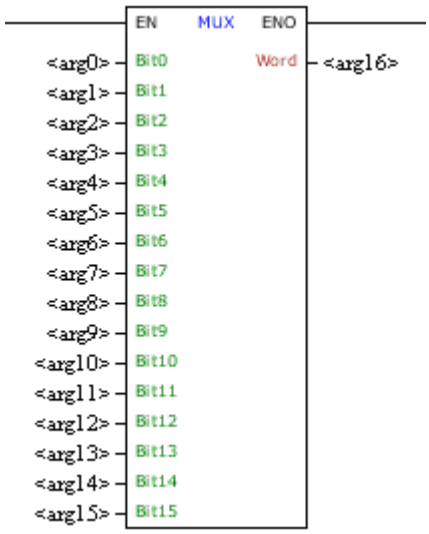
Em execução ("EN"=1), o conteúdo de "Value" , que é do tipo "DataType#", é transferido para a variável global localizada em "Address" do "Group#.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Group#"=HW_OUTPUT
"DataType#"=WORD
"Address"=6
"Value"=100. Novo valor da entrada analógica 1 (AI1).

10.1.1.19.6 MUX

Figura



Configuração do Bloco

<arg0>: "Bit0" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg1>: "Bit1" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg2>: "Bit2" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg3>: "Bit3" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg4>: "Bit4" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg5>: "Bit5" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg6>: "Bit6" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg7>: "Bit7" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg8>: "Bit8" - VAR_IN: inserir um número constante ou uma variável (tag). Tipo de dado: BOOL.

<arg9>: "Bit9" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: BOOL.

<arg10>: "Bit10" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg11>: "Bit11" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg12>: "Bit12" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg13>: "Bit13" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg14>: "Bit14" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg15>: "Bit15" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg16>: "Word" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.
Descrição: bits concatenados.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

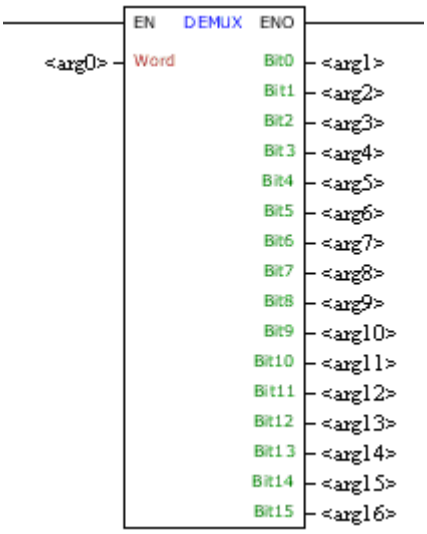
Em execução ("EN"=1), os valores "Bit0", "Bit1", "Bit2", ... e "Bit15" são transferidos para "Word"..

Se "EN" for 0, o valor em "Word" permanece inalterado.

Exemplo ("EN"=1)
"Bit0" = 1; "Bit1" = 0; "Bit2" = 0; "Bit3" = 0; "Bit4" = 1; "Bit5" = 1;
"Bit6" = 0; "Bit7" = 0; "Bit8" = 0; "Bit9" = 1; "Bit10" = 0; "Bit11" = 0;
"Bit12" = 0; "Bit13" = 0; "Bit14" = 1; "Bit15" = 1.
"Word"= -15823 = 16#1100_0010_0011_0001 (se INT).
"Word"= 49713 = 16#1100_0010_0011_0001 (se UINT).

10.1.1.19.7 DEMUX

Figura



Configuração do Bloco

<arg0>: "Word" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Bit0" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg2>: "Bit1" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg3>: "Bit2" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg4>: "Bit3" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg5>: "Bit4" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg6>: "Bit5" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg7>: "Bit6" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg8>: "Bit7" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg9>: "Bit8" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg10>: "Bit9" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg11>: "Bit10" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg12>: "Bit11" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg13>: "Bit12" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg14>: "Bit13" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg15>: "Bit14" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg16>: "Bit15" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o valor de "Word" são distribuído respectivamente ao "Bit0", "Bit1", "Bit2", ... e "Bit15".

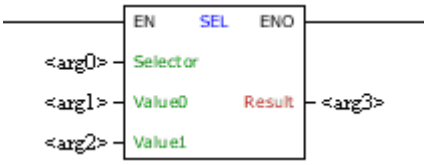
Se "EN" for 0, os valores em "Bit0", "Bit1", "Bit2", ... e "Bit15" permanecem inalterados.

Exemplo ("EN"=1)

"Word"= 49713 = 16#1100_0010_0011_0001.
"Bit0" = 1; "Bit1" = 0; "Bit2" = 0; "Bit3" = 0; "Bit4" = 1; "Bit5" = 1;
"Bit6" = 0; "Bit7" = 0; "Bit8" = 0; "Bit9" = 1; "Bit10" = 0; "Bit11" = 0;
"Bit12" = 0; "Bit13" = 0; "Bit14" = 1; "Bit15" = 1.

10.1.1.19.8 SEL

Figura



Configuração do Bloco

<arg0>: "Selector" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Value0" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg2>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg3>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), se "Selector"=0, "Result"="Value0" e se "Selector"=1, "Result"="Value1".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)
"Selector"=0.
"Value0" = 5.
"Value1" = 10.
"Result" = 5.

Exemplo2 ("EN"=1)
"Selector"=1.
"Value0" = 5.

"Value1" = 10.
"Result" = 10.

10.1.1.19.9 SWAP

Figura

Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT se "Value" for BYTE, USINT ou SINT.
Tipo de dado: WORD, UINT ou INT se "Value" for WORD, UINT ou INT.
Tipo de dado: DWORD, UDINT ou DINTse "Value" for DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito uma troca em "Value" e transferido ao "Result". O tamanho do tipo de dado de "Result" deve ser o mesmo que o de "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

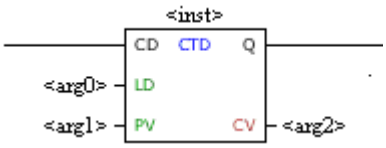
Exemplo1 ("EN"=1)
"Value" = 4660 = 16#1234
"Result" = 13330 = 16#3412

Exemplo2 ("EN"=1)
"Value" = 19088743 = 16#0123_4567
"Result" = 587294533 = 16#2301_6745

10.1.1.20 Counter

10.1.1.20.1 CTD

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "LD" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BOOL
Descrição: Load

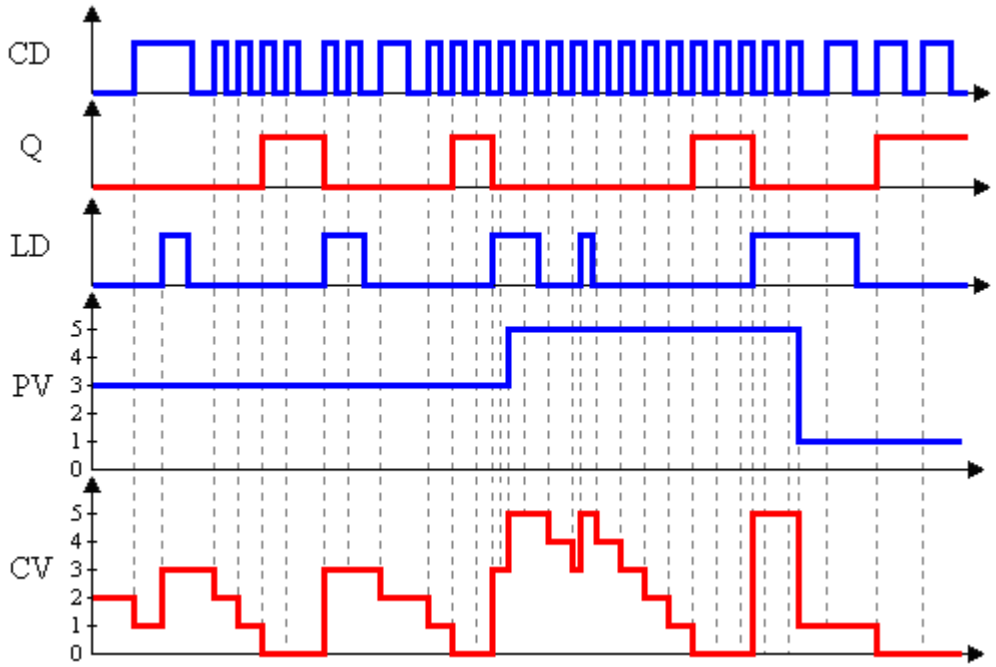
<arg1>: "PV" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Preset Value

<arg2>: "CV" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Counter Value
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

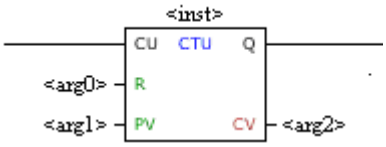
A cada pulso detectado em "CD", "CV" é decrementado. Quando "CV" for 0, o bloco carrega 1 para a saída "Q". Caso contrário, "Q"=0. Quando "LD"=1, "CV"= "PV".

Diagrama de Funcionamento



10.1.1.20.2 CTU

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "R" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BOOL
Descrição: Reset

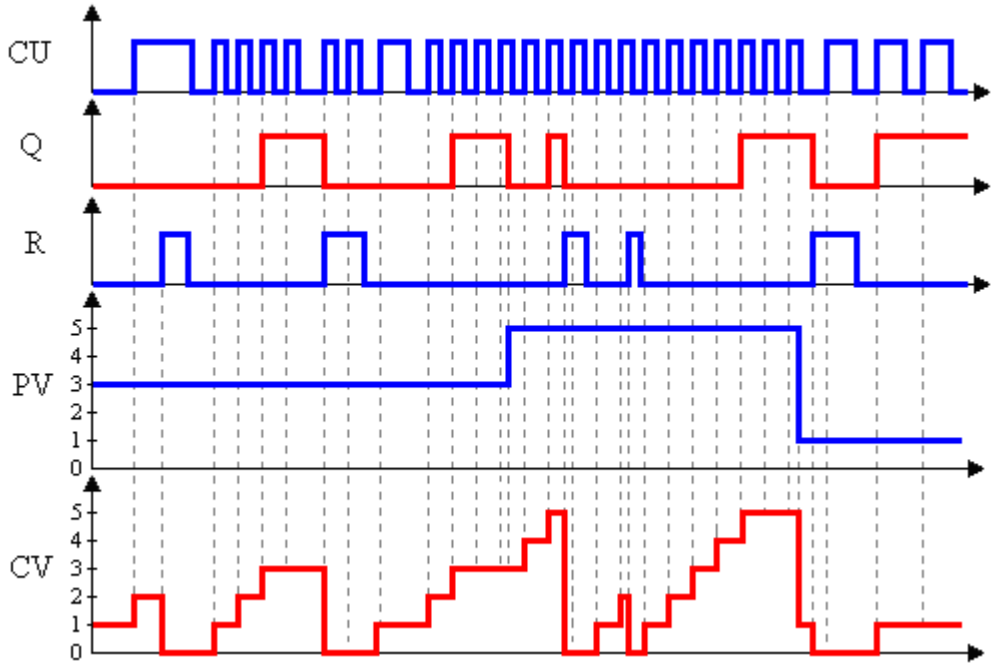
<arg1>: "PV" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Preset Value

<arg2>: "CV" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Counter Value
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

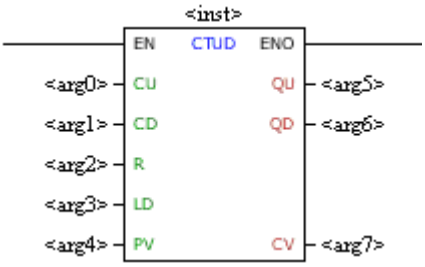
A cada pulso detectado em "CU", "CV" é incrementado. Quando "CV" atingir "PV, o bloco carrega 1 para a saída "Q". Caso contrário, "Q"=0. Quando "R"=1, "CV"=0.

Diagrama de Funcionamento



10.1.1.20.3 CTUD

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "CU" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Count Up

<arg1>: "CD" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BOOL

Descrição: Count Down

<arg2>: "R" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Reset

<arg3>: "R" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Load

<arg4>: "PV" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Preset Value

<arg5>: "QU" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Output Up
Nota: A variável tem que ter permissão de escrita.

<arg6>: "QD" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Output Down
Nota: A variável tem que ter permissão de escrita.

<arg7>: "CV" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Counter Value
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

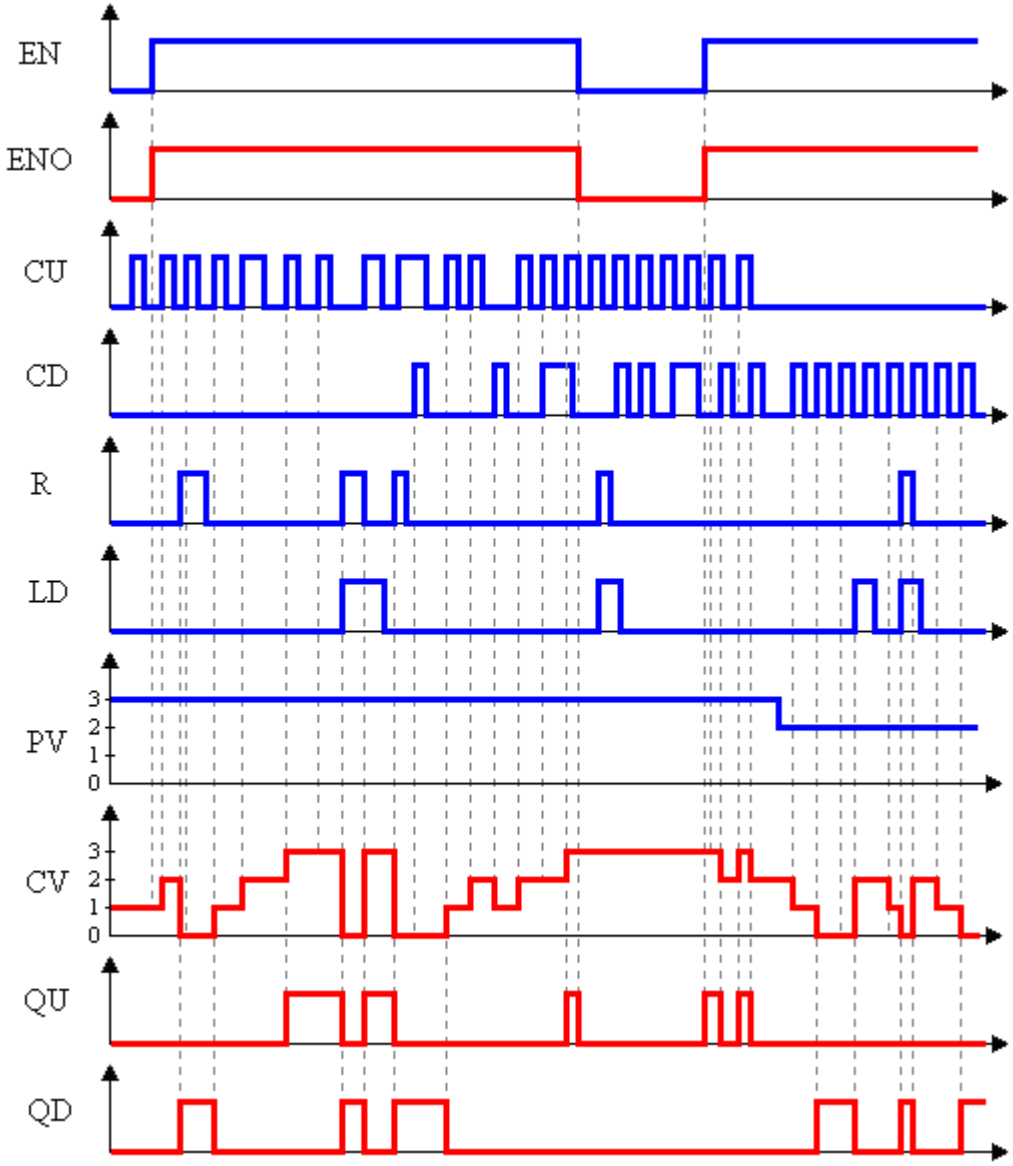
O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

A cada pulso detectado em "CU", "CV" é incrementado. Quando "CV" atingir "PV, o bloco carrega 1 para a saída "QU". Caso contrário, "QU"=0. Quando "R"=1, "CV"=0 e "QD"=1 .

Por outro lado, a cada pulso detectado em "CD", "CV" é decrementado. Quando "CV" for 0, o bloco carrega 1 para a saída "QD". Caso contrário, "QD"=0. Quando "LD"=1, "CV"= "PV" e "QU"=1.

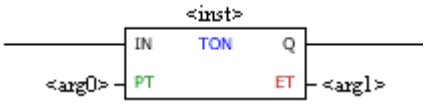
Diagrama de Funcionamento



10.1.1.21 Timer

10.1.1.21.1 TON

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "PT" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Preset Time [ms - milisegundos]

<arg1>: "ET" - VAR_IN: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Elapsed Timer [ms -milisegundos]
Nota: A variável tem que ter permissão de escrita.

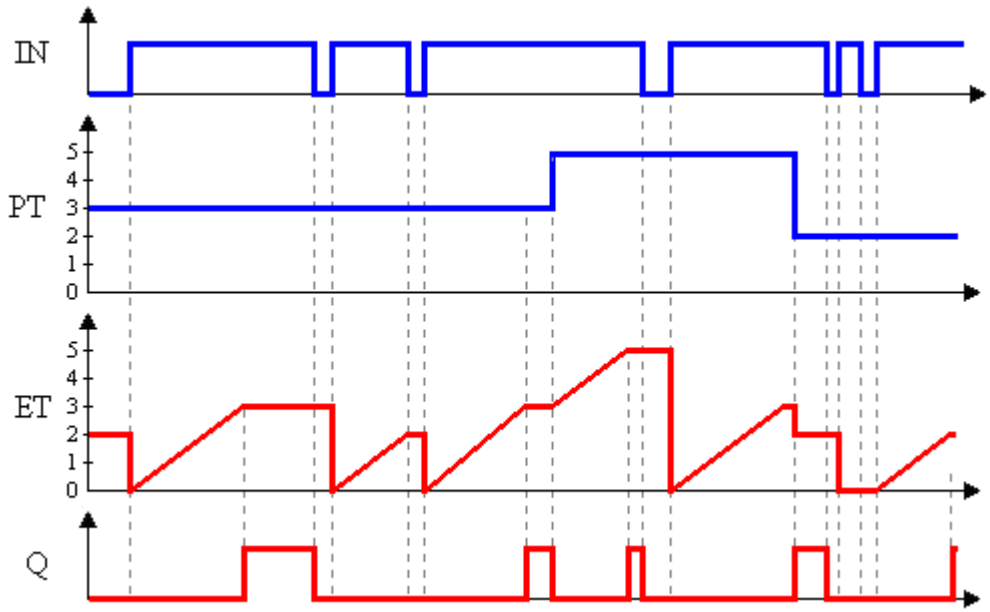
Descrição de Funcionamento

O bloco somente é executado quando a entrada "IN" for 1.

A saída "Q" vai para 0 enquanto "IN"=1 e "ET"<"PT". Após este tempo decorrido, a saída "Q" vai para 0, desde que a entrada "IN" permaneça em 1..

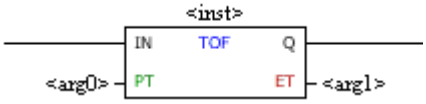
Se "IN"=0, "ET" permanece inalterado e a saída "Q"=0. Na borda de subida de "IN", "ET"=0.

Diagrama de Funcionamento



10.1.1.21.2 TOF

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "PT" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Preset Time [ms - milisegundos]

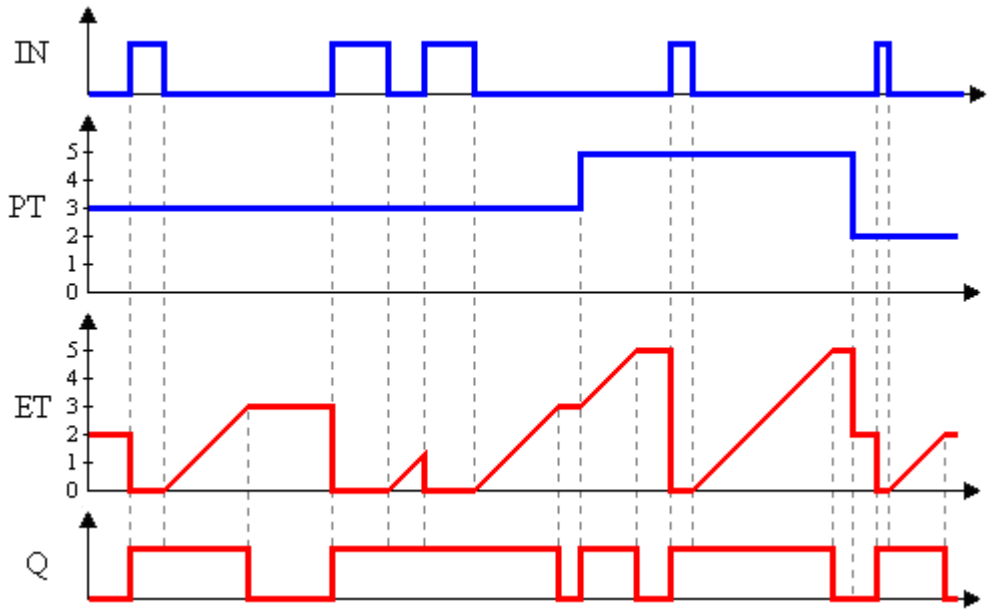
<arg1>: "ET" - VAR_IN: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Elapsed Timer [ms -milisegundos]
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco é iniciado na transição positiva de "IN" e a saída Q vai para 1.

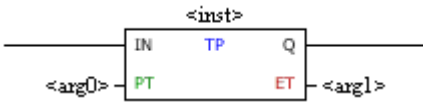
No instante em que "IN"=0, a temporização é iniciada e quando "ET" atingir "PT", a saída "Q"=0.

Diagrama de Funcionamento



10.1.1.21.3 TP

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "PT" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Preset Time [ms - milisegundos]

<arg1>: "ET" - VAR_IN: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Elapsed Timer [ms -milisegundos]
Nota: A variável tem que ter permissão de escrita.

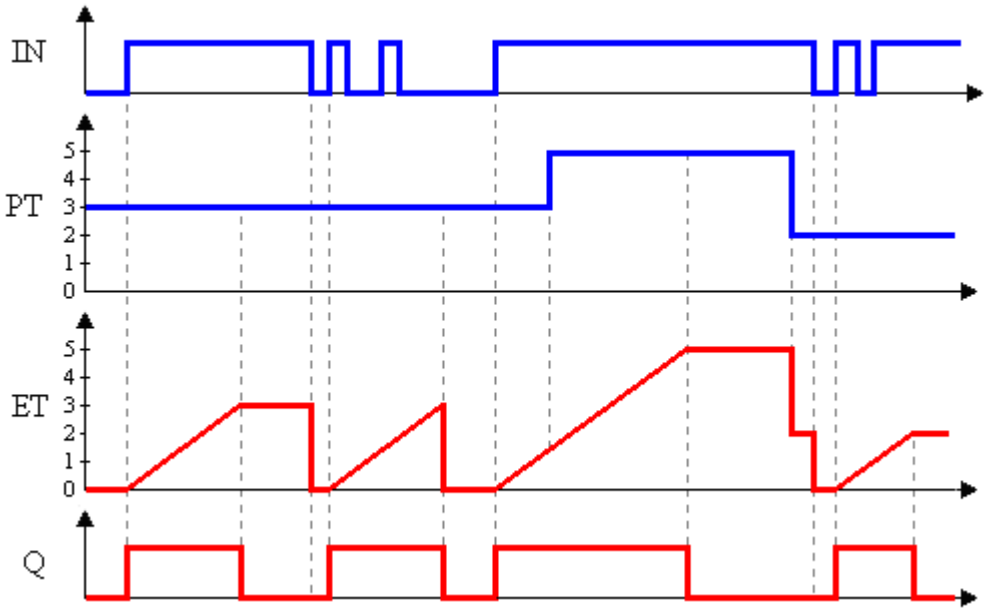
Descrição de Funcionamento

O bloco somente é executado quando ocorre uma borda de subida em "IN".

A saída "Q" vai para 1 enquanto "ET"<"PT". Após este tempo decorrido, a saída "Q" vai para 0.

Quando estiver em execução:
- Se "ET" atingir "PT" e se "IN" já estiver em 0, então "ET"=0
- Se "ET" atingir "PT" e se "IN" for 1, então, "ET"="PT".

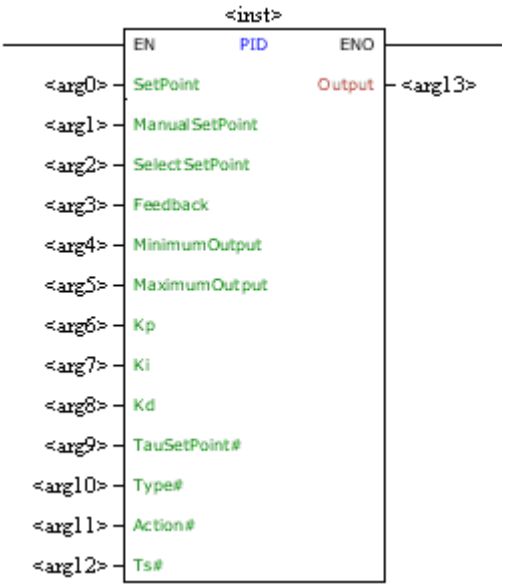
Diagrama de Funcionamento



10.1.1.22 Control

10.1.1.22.1 PID

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SetPoint" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: REAL
Descrição: Referência automática, ou seja, é o valor desejado para a variável de processo.

<arg1>: "ManualSetPoint" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL
Descrição: Referência manual, ou seja, é o valor transferido à saída do controlador.

<arg2>: "SelectorSetPoint" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: BOOL
Descrição: Seleciona a referência ativa: 0= "SetPoint"; 1= "ManualSetPoint".

<arg3>: "Feedback" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL
Descrição: Realimentação da saída do processo, ou seja, é a variável do processo.

<arg4>: "MinimumOutput" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL
Descrição: Valor mínimo permitido para a saída do controlador.

<arg5>: "MaximumOutput" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL

Descrição: Valor máximo permitido para a saída do controlador.

<arg6>: "Kp" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL
Descrição: Ganho proporcional do controlador, ou seja, é a correção proporcional ao erro.
Nota: A correção a ser aplicado ao processo deve crescer na proporção que cresce o erro entre o valor real (saída do processo) e desejado (referência).

<arg7>: "Ki" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL
Descrição: Ganho integral do controlador, ou seja, é a correção proporcional ao produto erro- tempo.
Nota: Erros pequenos mas que existem há muito tempo requerem correção mais intensa.

<arg8>: "Kd" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL
Descrição: Ganho derivativo do controlador, ou seja, é a correção proporcional à taxa de variação do erro em função do tempo.
Nota: Se o erro está variando muito rápido, esta taxa de variação deve ser reduzida para evitar oscilações.

<arg9>: "TauSetPoint#" - VAR_IN: inserir uma constante.
Tipos de dados: REAL
Descrição: Constante de tempo do filtro da referência automática.

<arg10>: "Type#" - VAR_IN: escolher uma das opções: "Academic"; "Paralel".
Tipos de dados: BYTE
Descrição: Tipo de controlador

<arg11>: "Action#" - VAR_IN: escolher uma das opções: "Direct"; "Reverse".
Tipos de dados: BYTE
Descrição: Ação de controle. Na ação direta, se a variável do processo aumenta, a saída do controlador também aumenta. Na ação reversa, se a variável do processo aumenta, a saída do controlador diminui.

<arg12>: "Ts#" - VAR_IN: inserir uma constante.
Tipos de dados: UINT
Descrição: Período de amostragem do controlador [ms].

<arg13>: "Output" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: REAL
Descrição: Saída do controlador que atua na entrada do processo.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

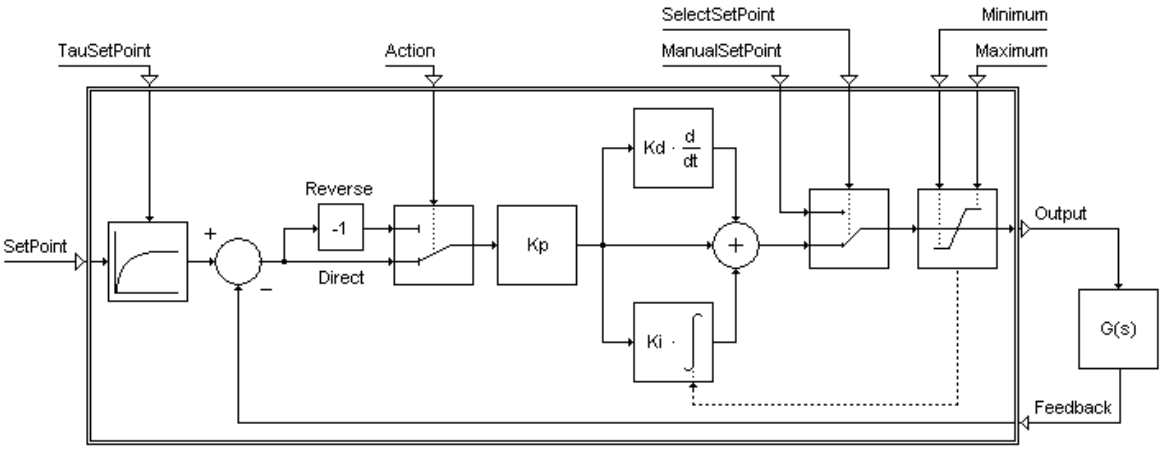
Se "EN"=0, "Output" permanece inalterado e a saída "ENO"=0. Na borda de subida de "EN", "Output"=0.

Diagrama de Blocos

- Acadêmico:

SPf: "SetPoint" após o filtro
Se "Action"="Direct", então: E(k)=SPf(k)-Feedback(k)
Se "Action"="Reverse", então: E(k)= Feedback(k)- SPf(k)

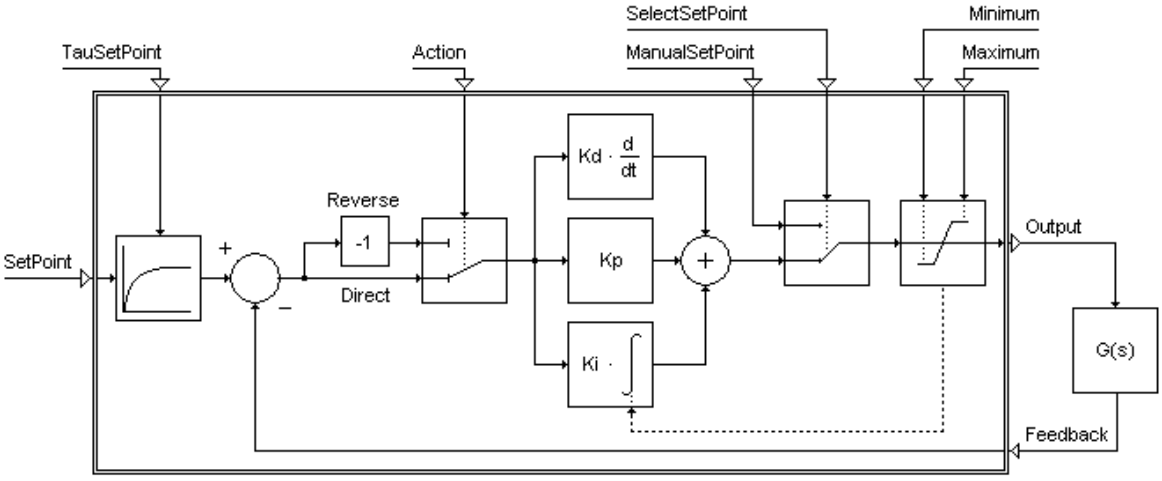
Output(k)=Output(k-1)+Kp*((1+Ki*Ts+(Kd/Ts))*e(k)-(Kd/Ts)*e(k-1))



- Paralelo:

SPf: "SetPoint" após o filtro
Se "Action"="Direct", então: E(k)=SPf(k)-Feedback(k)
Se "Action"="Reverse", então: E(k)= Feedback(k)- SPf(k)

Output(k)=Output(k-1)+(Kp+Ki*Ts+(Kd/Ts))*e(k)-(Kd/Ts)*e(k-1))



Efeito da alteração dos ganhos sobre o processo

- Se diminuir Kp, o processo torna-se mais lento; geralmente mais estável ou menos oscilante; tem menos sobre-sinal (overshoot).
- Se aumentar Kp, o processo responde mais rápido; pode ficar instável ou mais oscilante; tem mais sobre-sinal (overshoot).

- Se diminuir Ki, o processo torna-se mais lento, demorando para atingir o "SetPoint"; fica mais estável ou menos oscilante; tem menos sobre-sinal (overshoot).
- Se aumentar Ki, o processo torna-se mais rápido, atingindo rapidamente o "SetPoint"; fica mais instável ou mais oscilante; tem mais sobre-sinal (overshoot).

- Se diminuir Kd, o processo torna-se mais lento; tem menos sobre-sinal (overshoot).
- Se aumentar Kd, tem mais sobre-sinal (overshoot).

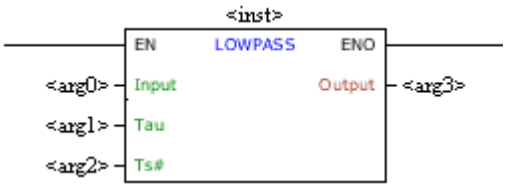
Como melhor o desempenho do processo pelo ajuste dos ganhos (válido para o PID Acadêmico)

- Se o desempenho do processo está quase bom, mas o sobre-sinal (overshoot) está um pouco alto, tente: (1) diminuir o Kp em 20%, (2) diminuir Ki em 20% e/ou (3) diminuir Kd em 50%.
- Se o desempenho do proceso está quase bom, mas não tem o sobre-sinal (overshoot) e demora para atingir o "SetPoint", tente: (1) aumentar Kp em 20%, (2) aumentar Ki em 20%, (3) aumentar KD em 50%.
- Se o desempenho do processo está bom, mas a saída do processo está variando demais, tente: (1) aumentar KD em 50%, (2) diminuir Kp em 20%.
- Se o desempenho do processo está ruim, ou seja, após a partida, o transitório dura vários períodos de oscilação, que reduz muito lentamente ou não reduz, tente: (1) diminuir Kp em 50%.
- Se o desempenho do processo está ruim, ou seja, após a partida avança lentamente em direção ao "SetPoint", sem sobre-sinal (overshoot), mas ainda está muito longe e a saída do processo é menor que o valor nominal, tente: (1) aumentar Kp em 50%, (2) aumentar Ki em 50%, (3) aumentar Ki em 70%.

10.1.1.23 Filter

10.1.1.23.1 LOWPASS

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Input" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: REAL
Descrição: Entrada do filtro.

<arg1>: "Tau" - VAR_IN: inserir uma constante ou variável (tag).
Tipos de dados: REAL
Descrição: Constante de tempo do filtro.

<arg2>: "Ts#" - VAR_IN: inserir uma constante.
Tipos de dados: UINT
Descrição: Período de amostragem do filtro [ms].

<arg3>: "Output" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: REAL
Descrição: Saída do filtro.
Nota: A variável tem que ter permissão de escrita.

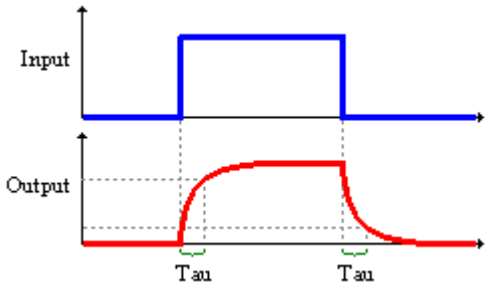
Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Se "EN"=0, "Output" permanece inalterado e a saída "ENO"=0. Na borda de subida de "EN", "Output"=0.

Diagrama de Funcionamento



10.1.1.24 Hardware

10.1.1.24.1 READENC

Figura

Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Source#" - VAR_IN: inserir uma das seguintes opções: "Quadrature_AB"; "Pulse_A"; "Pulse_B"; "Pulse_Z".
Tipos de dados: BYTE
Descrição: Fonte de contagem

<arg1>: "Value" - VAR_OUTN: inserir uma variável (tag).
Tipos de dados: DWORD ou UDINT
Descrição: Valor de contagem
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito a leitura do número de pulsos, conforme "Source" e transferidos para "Value". A cada transição de subida em "EN", o conteúdo de "Value" é inicializado em 0.

Se "EN" for 0, o valor em "Value" permanece inalterado.

10.1.1.24.2 READENC2

Figura

Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Source#" - VAR_IN: inserir uma das seguintes opções: "Quadrature_AB"; "Pulse_A"; "Pulse_B"; "Pulse_Z".
Tipos de dados: BYTE
Descrição: Fonte de contagem

<arg1>: "Pulses#" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: DWORD ou UDINT
Descrição: Número de pulsos para reiniciar contagem.

<arg2>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: DWORD ou UDINT
Descrição: Valor de contagem
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito a leitura do número de pulsos, conforme "Source" e transferidos para "Value". A cada transição de subida em "EN", o conteúdo de "Value" é inicializado em 0. Além disso, toda vez que a contagem for crescendo e "Value" atingir "Pulses", "Value" é reiniciado para 0. Porém, se a contagem decrescer, ou seja, "Value" tender a ficar menor que 0, então "Value" vai para "Pulses".

Se "EN" for 0, o valor em "Value" permanece inalterado.

10.1.1.24.3 READENC3

Figura

Configuração do Bloco

<arg0>: "Source#" - VAR_IN: selecionar uma das seguintes opções: "Quadrature_AB"; "Pulse_A"; "Pulse_B"; "Pulse_Z".
Tipos de dados: BYTE
Descrição: Valor da contagem.

<arg1>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: DWORD ou UDINT
Descrição: Número de pulsos contados pelo PLC300.
Nota: A variável tem que ter permissão de escrita.

<arg2>: "Delta" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: DINT
Descrição: Diferença entre o número de pulsos contados nessa execução do bloco com a execução anterior.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feita a leitura do número de pulsos contados desde quando o PLC300 foi energizado, conforme "Source#" e transferidos para "Value". O valor de

"Delta" será a diferença entre os pulsos contados nessa execução do bloco com a execução anterior, ou seja, se o bloco foi executado por uma TASK de evento externo DI9 e depois executado por uma TASK de evento externo DI10, o valor de "Delta" será o número de pulsos contados entre o acionamento da DI9 e DI10.

Se "EN" for 0, o valor em "Value" e "Delta" permanecem inalterados.

10.1.1.24.4 PWM

Figura

Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Frequency" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: DWORD ou UDINT
Descrição: Freqüência do PWM - 0 a 300000 Hz

<arg1>: "Value" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Largura do pulso - 0 a 1000 (onde 1000 equivale a 100%)

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

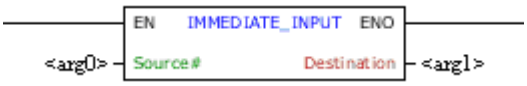
Pelo fato deste bloco utilizar a saída digital 9 (DO9), quando em execução ("EN"=1), é escrito nesta saída um PWM conforme a "Frequency" e "Width".

IMPORTANTE: Este bloco ativo ("EN"=1) tem prioridade de comando em relação a qualquer bobina que tente comandar a saída digital 9.

Se "EN" for 0, o valor em "Value" permanece inalterado.

10.1.1.24.5 IMMEDIATE_INPUT

Figura



Configuração do Bloco

<arg0>: "Source#" - VAR_IN: inserir uma das seguintes opções: "DI01... DI10"; "AI01".
Tipos de dados: BYTE
Descrição: Origem dos dados (DI1 à DI10 ou AI1).

<arg1>: "Destination" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD, INT ou UINT
Descrição: Valor instantâneo da(s) entrada(s).
Nota: A variável tem que ter permissão de escrita.

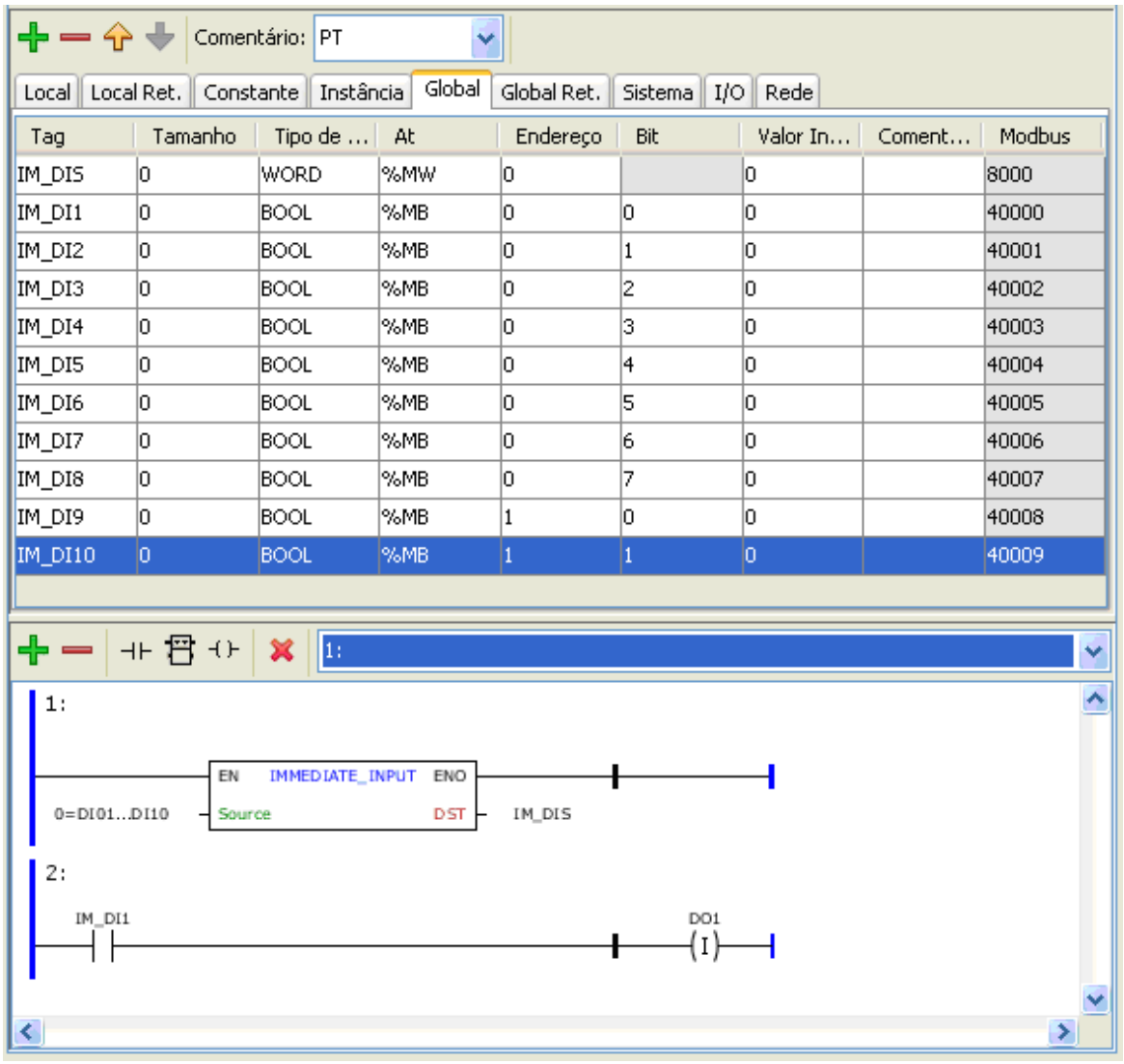
Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito a leitura instantânea das entradas DI1 à DI10 ou AI1, conforme "Source#" e transferidos para "Destination", sem alterar as respectivas imagens (entradas digitais ou entrada analógica). As imagens somente são atualizadas no início do próximo SCAN do programa.

Para ler uma entrada digital individualmente, basta criar uma variável no mesmo endereço da variável "arg1" e programar o bit desejado. No exemplo abaixo, "IM_DIS" é a variável "arg1"(WORD) e os bits IM_DI1 a IM_DI10, são booleanos mapeados sobre a IM_DIS.


Comentário: PT
Local
Local Ret.
Constante
Instância
Global
Global Ret.
Sistema
I/O
Rede
Tag | Tamanho | Tipo de ... | At | Endereço | Bit | Valor In... | Coment... | Modbus
IM_DIS | 0 | WORD | %MW | 0 | | 0 | | 8000
IM_DI1 | 0 | BOOL | %MB | 0 | 0 | 0 | | 40000
IM_DI2 | 0 | BOOL | %MB | 0 | 1 | 0 | | 40001
IM_DI3 | 0 | BOOL | %MB | 0 | 2 | 0 | | 40002
IM_DI4 | 0 | BOOL | %MB | 0 | 3 | 0 | | 40003
IM_DI5 | 0 | BOOL | %MB | 0 | 4 | 0 | | 40004
IM_DI6 | 0 | BOOL | %MB | 0 | 5 | 0 | | 40005
IM_DI7 | 0 | BOOL | %MB | 0 | 6 | 0 | | 40006
IM_DI8 | 0 | BOOL | %MB | 0 | 7 | 0 | | 40007
IM_DI9 | 0 | BOOL | %MB | 1 | 0 | 0 | | 40008
IM_DI10 | 0 | BOOL | %MB | 1 | 1 | 0 | | 40009
1:
EN
IMMEDIATE_INPUT
ENO
0=DI01...DI10
Source
DST
IM_DIS
2:
IM_DI1
DO1

Se "EN" for 0, o valor em "Destination" permanece inalterado.

10.1.1.24.6 IMMEDIATE_OUTPUT

Figura

Configuração do Bloco

<arg0>: "Source" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD, INT ou UINT
Descrição: Valor de escrita.

<arg1>: "Destination#" - VAR_OUT: inserir uma das seguintes opções: "DO01 à DO9" ou "AO1".
Tipos de dados: BYTE
Descrição: Destino de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

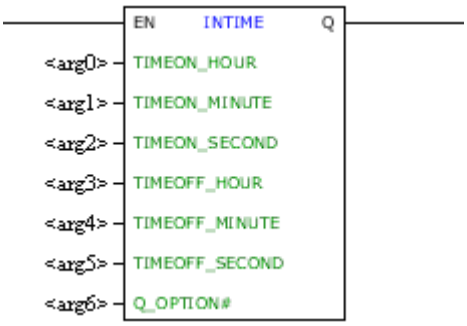
A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o valor de "Source" é transferido imediatamente para as saídas digitais DO01 à DO9 ou AO1, conforme "Destination#, sem aguardar o fim do SCAN do programa. A imagem da saída selecionada também é alterada.

10.1.1.25 RTC

10.1.1.25.1 INTIME

Figura



Configuração do Bloco

<arg0>: "TIMEON_HOUR" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Hora para ligar

<arg1>: "TIMEON_MINUTE" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Minuto para ligar.

<arg2>: "TIMEON_SECOND" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Segundo para ligar

<arg3>: "TIMEOFF_HOUR" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Hora para desligar

<arg4>: "TIMEOFF_MINUTE" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT
Descrição: Minuto para desligar.

<arg5>: "TIMEOFF_SECOND" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Segundo para desligar

<arg6>: "Q_OPTION#" - VAR_IN: inserir uma das seguintes opções: Normal; Invertido
Tipos de dados: BYTE
Descrição: Opção de funcionamento da saída "Q".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1) e se "Q_OPTION"=Normal, a saída "Q" somente vai a 1 quando a hora atual do relógio for maior ou igual ao horário configurado em "TIMEON_HOUR", "TIMEON_MINUTE" e "TIMEON_SECOND" e menor que o horário configurado em "TIMEOFF_HOUR", "TIMEOFF_MINUTE" e "TIMEOFF_SECOND". Senão, a saída "Q" fica em 0.

Em execução ("EN"=1) e se "Q_OPTION"=Inverted, então o funcionamento da saída "Q" é o oposto da descrição do parágrafo anterior.

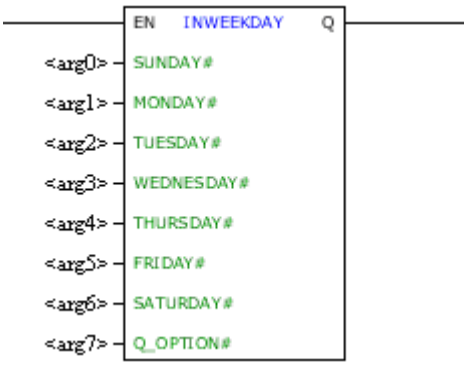
Se "EN" for 0 então a saída "Q" também será 0.

Exemplo ("EN"=1 e "Q_OPTION#"=Normal)
"TIMEON_HOUR"=14
"TIMEON_MINUTE"=00
"TIMEON_SECOND"=00
"TIMEOFF_HOUR"=16
"TIMEOFF_MINUTE"=00
"TIMEOFF_SECOND"=00
Se a hora corrente=13:59:59, então "Q"=0
Se a hora corrente=14:00:00, então "Q"=1
Se a hora corrente=16:00:00, então "Q"=0

Exemplo ("EN"=1 e "Q_OPTION#"=Inverted)
"TIMEON_HOUR"=14
"TIMEON_MINUTE"=00
"TIMEON_SECOND"=00
"TIMEOFF_HOUR"=16
"TIMEOFF_MINUTE"=00
"TIMEOFF_SECOND"=00
Se a hora corrente=13:59:59, então "Q"=1
Se a hora corrente=14:00:00, então "Q"=0
Se a hora corrente=16:00:00, então "Q"=1

10.1.1.25.2 INWEEKDAY

Figura



Configuração do Bloco

<arg0>: "SUNDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg1>: "MONDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg2>: "TUESDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg3>: "WEDNESDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg4>: "THURSDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg5>: "FRIDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg6>: "SATURDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg7>: "Q_OPTION#" - VAR_IN: inserir uma das seguintes opções: Normal; Invertido
Tipos de dados: BYTE
Descrição: Opção de funcionamento da saída "Q".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1) e se "Q_OPTION"=Normal, a saída "Q" somente vai a 1 quando o Dia da Semana: "SUNDAY#", "MONDAY#", "TUESDAY#", "WEDNESDAY#", "THURSDAY#", "FRIDAY#" ou "SATURDAY#" estiver "Enabled" e o dia da semana corrente do relógio for o mesmo. Senão, a saída "Q" fica em 0.

Em execução ("EN"=1) e se "Q_OPTION"=Inverted, então o funcionamento da saída "Q" é o oposto da descrição do parágrafo anterior.

Se "EN" for 0 então a saída "Q" também será 0.

Exemplo1 ("EN"=1 e "Q_OPTION"=Normal)
"SUNDAY#"=0
"MONDAY#"=1
"TUESDAY#"=1
"WEDNESDAY#"=1
"THURSDAY#"=1
"FRIDAY#"=1
"SATURDAY#"=0
Se o dia corrente =Domingo, então "Q"=0
Se o dia corrente =Terça, então "Q"=1
Se o dia corrente =Sábado, então "Q"=0

Exemplo2 ("EN"=1 e "Q_OPTION"=Inverted)
"SUNDAY#"=0
"MONDAY#"=1
"TUESDAY#"=1
"WEDNESDAY#"=1
"THURSDAY#"=1
"FRIDAY#"=1
"SATURDAY#"=1
Se o dia corrente =Domingo, então "Q"=1
Se o dia corrente =Terça, então "Q"=0
Se o dia corrente =Sábado, então "Q"=1

10.1.1.26 Screen

10.1.1.26.1 SETSCREEN

Figura

Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "NUMBER" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Número da Tela a ser mostrada na HMI.

Descrição de Funcionamento

O bloco somente é executado na transição de subida da entrada "EN".

Na transição de subida de EN, a saída "Q" vai para 1 somente por 1 ciclo de SCAN, desde que a número da tela seja válido. Em qualquer outra condição, a saída "Q" fica em 0.

Nota: Ao utilizar este bloco, verificar a necessidade ou não de ter senha na TELA configurada. Caso a senha tenha sido programada, ao mudar para a TELA utilizando este bloco, o sistema irá solicitar a devida senha.

10.1.1.27 CANopen

10.1.1.27.1 CANopen

Operação na Rede CANopen - Modo Mestre

Além da operação como escravo, o controlador programável PLC300 também permite a operação como mestre da rede CANopen. A seguir serão descritas características e funções do PLC300 como mestre da rede CANopen.

Habilitação da Função CANopen Mestre

Como padrão, o controlador programável PLC300 está programado para operar como escravo da rede CANopen. A programação do equipamento como mestre da rede deve ser feita utilizando o software WSCAN, que permite também a configuração de toda a rede CANopen. A descrição detalhada das janelas e funções do software WSCAN deve ser obtida no menu "Ajuda" do próprio software.

Depois de elaborada a configuração do mestre, é necessário fazer o download das configurações, utilizando uma das interfaces de programação do produto - consulte o manual do usuário para maiores informações. Uma vez programado como mestre da rede, caso seja necessário apagar estas configurações, a função para apagar o programa do usuário - disponível no menu Setup - também apaga as configurações do mestre CANopen.

Nota: A rede CANopen é uma rede flexível e que permite várias formas de configuração e operação. No entanto, esta flexibilidade exige que o usuário tenha bom conhecimento das funções e objetos de comunicação utilizados para configuração da rede, bem como o conhecimento do software de programação WSCAN.

Características do Mestre CANopen

O controlador programável PLC300 permite controlar um conjunto de até 63 escravos, utilizando os seguintes serviços e recursos de comunicação:

- Serviço de gerenciador da rede (NMT)
- 63 PDOs de transmissão
- 63 PDOs de recepção
- 63 Consumidores Heartbeat
- Produtor Heartbeat
- Cliente SDO
- Produtor/consumidor SYNC
- 512 bytes de marcadores de rede de entrada (input)
- 512 bytes de marcadores de rede de saída (output)

As características físicas - instalação, conector, cabo, etc. - são as mesmas, tanto para o PLC300 operando como mestre quanto como escravo. As configurações de endereço e taxa de comunicação também são necessárias para a operação como mestre, mas estas configurações são programadas pelo software WSCAN de acordo com as propriedades definidas para o mestre no próprio software.

Nota: Os marcadores de rede de entrada são utilizados para mapear dados nos RPDOs, enquanto que os marcadores de rede de saída são utilizados para mapear dados nos TPDOs. Eles podem ser acessados em Byte (%IB ou %QB), Word (%IW ou %QW) ou Double Word (%ID ou %QD). Sua função, no entanto, não é pré-definida, e depende do aplicativo em ladder desenvolvido para o controlador PLC300.

Operação do Mestre

Uma vez programado para operar como mestre, o controlador programável PLC300 executará as seguintes etapas para realizar a inicialização, em seqüência, para cada um dos escravos:

- 1 - Enviado o comando de reset da comunicação para toda a rede, para que os escravos iniciem com valores conhecidos para os objetos de comunicação.
- 2 - Identificação do equipamento na rede, através da leitura via SDO do objeto 1000h/00h - Object Identification.
- 3 - Escrita via SDO de todos os objetos programados para o escravo, que usualmente inclui a configuração e mapeamento dos TPDOs e RPDOs, node guarding, heartbeat, além dos objetos específicos do fabricante, caso sejam programados.
- 4 - Iniciado serviço de controle de erros - node guarding ou heartbeat - caso sejam programados.
- 5 - Envio do escravo para modo operacional.

Se uma destas etapas falhar, será indicado erro de comunicação com o escravo. Dependendo das configurações, a inicialização dos escravos será abortada, e o mestre fará a inicialização do escravo seguinte, retornando para o escravo com erro após tentar inicializar todos os demais escravos da rede.

De forma semelhante, se, durante a operação de um escravo, for identificado erro no serviço de controle de erros, dependendo das configurações feitas para o mestre, o

escravo será automaticamente resetado e o procedimento de inicialização será executado novamente.

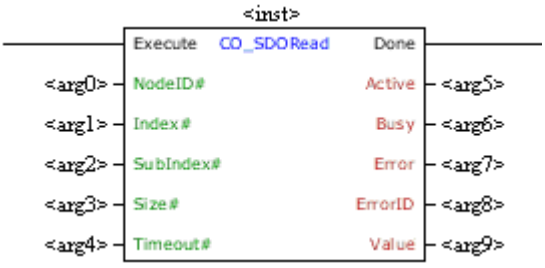
Nota: O estado da comunicação e o estado de cada escravo podem ser observados em marcadores de sistema de entrada.

Blocos para o Mestre CANopen

Além dos objetos de comunicação e das configurações feitas no software WSCAN, também estão disponíveis blocos para monitoração e envio de comandos, que podem ser utilizados durante a elaboração do aplicativo em ladder para o controlador programável PLC300. Não é necessário utilizar estes blocos durante a operação do equipamento, mas seu uso confere maior flexibilidade e facilita o diagnóstico de problemas de comunicação durante a operação do controlador programável PLC300.

10.1.1.27.2 CO_SDORead

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "NodeID#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Endereço do escravo destino - 1 a 127.

<arg1>: "Index#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Índice do objeto acessado, dentre os objetos disponíveis no dicionário de objetos do escravo - 0 a 65535.

<arg2>: "SubIndex#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Sub-índice do objeto acessado - 0 a 255.

<arg3>: "Size#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Tamanho do dado acessado, em bytes - 1 a 4.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD
Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 5 a 5000 ms.

<arg5>: "Active" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco ativo, requisição de leitura enviada para o escravo e aguardando resposta.
Nota: A variável tem que ter permissão de escrita.

<arg6>: "Busy" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco habilitado, mas recurso não está disponível (cliente SDO enviando outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.
Nota: A variável tem que ter permissão de escrita.

<arg7>: "Error" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Erro na execução da requisição.
Nota: A variável tem que ter permissão de escrita.

<arg8>: "ErrorID" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Cartão não pode executar a função" (exemplo: mestre não habilitado); 2= "Timeout na resposta do escravo"; 3= "Escravo retornou erro".
Nota: A variável tem que ter permissão de escrita.

<arg9>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE[1 ... 4] ou USINT[1 ... 4]
Descrição: Variável ou array onde serão salvos os dados lidos do escravo.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para leitura de dados via SDO de um escravo remoto. Permite realizar a leitura de objetos na rede com tamanho de até 4 bytes.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução com sucesso da função. Na transição positiva de "Execute", quando o cliente SDO do mestre estiver livre, uma nova requisição é enviada para o servidor SDO do escravo. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

10.1.1.27.3 CO_SDOWrite

Figura

Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "NodeID#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Endereço do escravo destino - 1 a 127.

<arg1>: "Index#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Índice do objeto acessado, dentre os objetos disponíveis no dicionário de objetos do escravo - 0 a 65535.

<arg2>: "SubIndex#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Sub-índice do objeto acessado - 0 a 255.

<arg3>: "Size#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Tamanho do dado acessado, em bytes - 1 a 4.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 5 a 5000 ms.

<arg5>: "Value" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BYTE[1 ... 4] ou USINT[1 ... 4]
Descrição: Variável ou array com dados para enviar para o escravo.

<arg6>: "Active" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco ativo, requisição de escrita enviada para o escravo e aguardando resposta.
Nota: A variável tem que ter permissão de escrita.

<arg7>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL
Descrição: Bloco habilitado, mas recurso não está disponível (cliente SDO enviando outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.
Nota: A variável tem que ter permissão de escrita.

<arg8>: "Error" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Erro na execução da requisição.
Nota: A variável tem que ter permissão de escrita.

<arg9>: "ErrorID" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Cartão não pode executar a função" (exemplo: mestre não habilitado); 2= "Timeout na resposta do escravo"; 3= "Escravo retornou erro".
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de dados via SDO para um escravo remoto. Permite realizar a escrita de objetos na rede com tamanho de até 4 bytes.

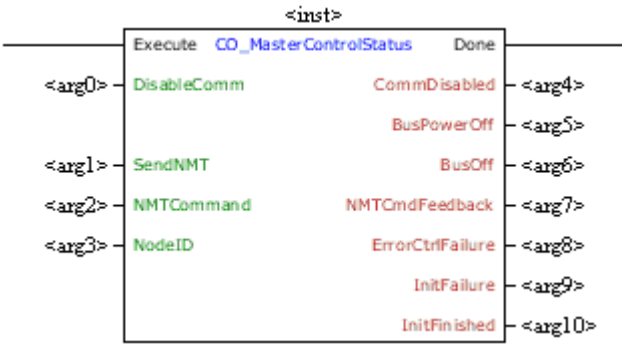
Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução com sucesso da função. Na transição positiva de "Execute", quando o cliente SDO do mestre estiver livre, uma nova requisição é enviada para o servidor SDO do escravo. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Nota

- É importante que a quantidade de dados lidos ou escritos programado nos blocos seja compatível com o tamanho da variável ou do array com o valor,
- Em caso de erro retornado pelo escravo, é possível obter o código do último erro recebido através dos marcadores de sistema de leitura. Consulte o item "Marcadores de Sistema do CAN" para a lista de marcadores disponíveis.

10.1.1.27.4 CO_MasterControlStatus

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "DisableComm" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BOOL
Descrição: Desabilita a comunicação CANopen. Ao desabilitar o mestre, os contadores e marcadores de status do mestre CANopem também são zerados - 0 ou 1.

<arg1>: "SendNMT" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BOOL
Descrição: Na transição de deste sinal o mestre CANopen dispara o envio de um comando de gerenciamento - NMT - de acordo com o comando e o endereço programado neste bloco - 0 ou 1.

<arg2>: "NMTCommand" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE
Descrição: Indica qual comando deve ser enviado para o escravo: 1= "Start node"; 2= "Stop node"; 128= "Enter pre-operational"; 129= "Reset node"; 130= "Reset communication".

<arg3>: "NodeID" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Endereço do escravo para envio do comando NMT - 0= Broadcast (mensagem para todos os escravos); 1 a 127= Endereço específico do escravo.

<arg4>: "CommDisabled" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre e a comunicação na interface CAN foi desabilitada. É indicado sempre que o comando do usuário para desabilitar a interface for recebido, mas também é indicado nas situações de falta de alimentação na interface CAN ou bus off: 0= "Comunicação habilitada"; 1= "Comunicação desabilitada".
Nota: A variável tem que ter permissão de escrita.

<arg5>: "BusPowerOff" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL

Descrição: Indica que foi detectado falha na alimentação da interface CAN: 0= "Interface CAN alimentada"; 1= "Interface CAN sem alimentação".
Nota: A variável tem que ter permissão de escrita.

<arg6>: "BusOff" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que foi detectado erro de bus off na interface CAN: 0= "Sem erro de bus off"; 1= "Com erro de bus off".
Nota: A variável tem que ter permissão de escrita.

<arg7>: "NMTCmdFeedback" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o comando de gerenciamento foi enviado pelo mestre: 0= "Sem comando ou comando não enviado"; 1= "Comando NMT enviado".
Nota: A variável tem que ter permissão de escrita.

<arg8>: "ErrorCtrlFailure" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre detectou erro no serviço de controle de erros (node guarding ou heartbeat) em pelo menos um escravo da rede: 0= "Sem erro detectado"; 1= "Mestre detectou erro de node guarding ou heartbeat em pelo menos um escravo da rede".
Nota: A variável tem que ter permissão de escrita.

<arg9>: "InitFailure" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre detectou erro durante a inicialização de pelo menos um escravo da rede: 0= "Sem erro detectado"; 1= "Mestre detectou erro na inicialização em pelo menos um escravo da rede".
Nota: A variável tem que ter permissão de escrita.

<arg10>: "InitFinished" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre tentou fazer a inicialização de todos os escravos da rede. Não necessariamente a inicialização foi executada com sucesso, podem ter ocorrido erros durante a inicialização: 0= "Mestre ainda não executou o procedimento de inicialização de todos os escravos"; 1= "Mestre fez a inicialização (com ou sem sucesso) de todos os escravos".
Nota: A variável tem que ter permissão de escrita.

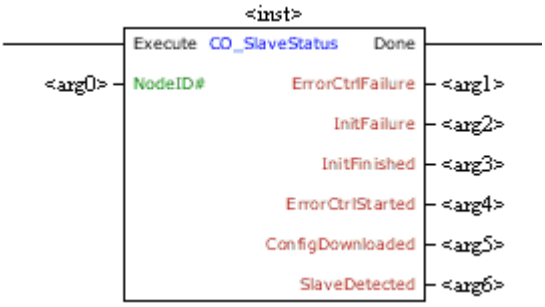
Descrição de Funcionamento

Bloco para controle e monitoração do mestre da rede CANopen. Mostra o estado do mestre da rede para diagnóstico e identificação de problemas de comunicação, bem como permite o envio de comandos para os serviços de gerenciamento da rede - NMT.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Se a entrada "Execute" está ativa, ele atualiza os valores das entradas e saídas e ativa a saída "Done". Se a entrada "Execute" não estiver ativa, os demais valores das entradas são ignorados e todas as saídas são zeradas.

10.1.1.27.5 CO_SlaveStatus

Figura



Configuração do Bloco

<arg0>: "NodeID" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Endereço do escravo para identificação do estado da comunicação com o mestre - 1 a 127.

<arg1>: "ErrorCtrlFailure" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre detectou erro no serviço de controle de erros (node guarding ou heartbeat) no escravo indicado: 0= "Sem erro detectado"; 1= "Mestre detectou erro de node guarding ou heartbeat no escravo".
Nota: A variável tem que ter permissão de escrita.

<arg2>: "InitFailure" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre detectou erro durante a inicialização do escravo indicado: 0= "Sem erro detectado"; 1= "Mestre detectou erro na inicialização no escravo".
Nota: A variável tem que ter permissão de escrita.

<arg3>: "InitFinished" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre fez a inicialização completa e com sucesso do escravo indicado: 0= "Mestre não concluiu o procedimento de inicialização do escravo"; 1= "Mestre fez a inicialização do escravo com sucesso".
Nota: A variável tem que ter permissão de escrita.

<arg4>: "ErrprCtrlStarted" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre iniciou o serviço de controle de erros (node guarding ou heartbeat) com o escravo indicado. Se este serviço não for habilitado para o escravo, este bit será ativado após realizar a configuração: 0= "Controle de erros com o escravo não iniciado"; 1= "Controle de erros com o escravo iniciado".
Nota: A variável tem que ter permissão de escrita.

<arg5>: "ConfigDownloaded" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre finalizou com sucesso o download das configurações via SDO para o escravo indicado: 0= "Mestre não finalizou o download das configurações para o escravo"; 1= "Download das configurações para o escravo finalizado com sucesso".
Nota: A variável tem que ter permissão de escrita.

<arg6>: "SlaveDetected" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica que o mestre conseguiu fazer a leitura da identificação via SDO do escravo indicado: 0= "Escravo não foi contactado"; 1= "Escravo contactado com sucesso".
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para monitoração do escravo da rede CANopen. Mostra o estado do escravo da rede para diagnóstico e identificação de problemas de comunicação.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Se a entrada "Execute" está ativa, ele atualiza os valores das entradas e saídas e ativa a saída "Done". Se a entrada "Execute" não estiver ativa, os demais valores das entradas são ignorados e todas as saídas são zeradas.

Nota

- Os dados acessados utilizando este bloco também estão disponíveis através de marcadores de sistema de leitura e escrita, conforme descrito no item "Marcadores de Sistema do CAN".

10.1.1.28 Modbus

10.1.1.28.1 Modbus

Operação na Rede Modbus RTU - Modo mestre

Além da operação como escravo, o controlador programável PLC300 também permite a operação como mestre da rede Modbus RTU. Para esta operação, é necessário observar os seguintes pontos:

- Somente a interface RS485 permite operação como mestre da rede.
- É necessário programar, nas configurações do produto, o modo de operação como "Mestre", além da taxa de comunicação, paridade e stop bits, que devem ser as mesmas para todos os equipamentos da rede.
- O mestre da rede Modbus RTU não possui endereço, logo o endereço configurado no PLC300 não é utilizado.
- O envio e recepção de telegramas via interface RS485 utilizando o protocolo Modbus RTU é programado utilizando blocos em linguagem de programação ladder.

É necessário conhecer os blocos disponíveis e o software de programação em ladder para poder programar o mestre da rede.

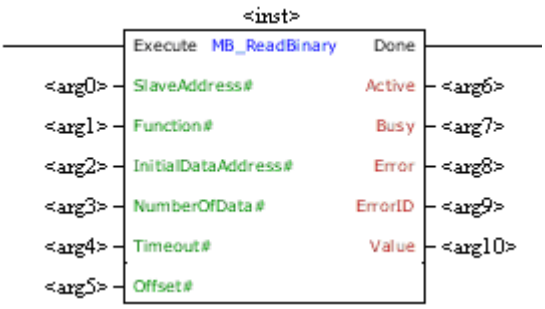
- As seguintes funções estão disponíveis para envio de requisições pelo mestre Modbus:
  - Função 01: Read Coils
  - Função 02: Read Discrete Inputs
  - Função 03: Read Holding Registers
  - Função 04: Read Input Registers
  - Função 05: Write Single Coil
  - Função 06: Write Single Register
  - Função 15: Write Multiple Coils
  - Função 16: Write Multiple Registers

Blocos para a programação do mestre

Para o controle e monitoração da comunicação Modbus RTU utilizando o controlador programável PLC300, foram desenvolvidos os seguintes blocos, que devem ser utilizados durante a programação em ladder.

10.1.1.28.2 MB_ReadBinary

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Código da função de leitura: 1= "Read Coils"; 2= "Read Discrete Inputs".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Endereço do bit inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE
Descrição: Número de bits lidos em sequência a partir do endereço inicial - 1 a 128.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.
Tipos de dados: BOOL
Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg6>: "Active" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco ativo, requisição de leitura enviada para o escravo e aguardando resposta.
Nota: A variável tem que ter permissão de escrita.

<arg7>: "Busy" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.
Nota: A variável tem que ter permissão de escrita.

<arg8>: "Error" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Erro na execução da requisição.
Nota: A variável tem que ter permissão de escrita.

<arg9>: "ErrorID" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".
Nota: A variável tem que ter permissão de escrita.

<arg10>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL[1 ... 128]
Descrição: Variável ou array onde serão salvos os dados lidos do escravo.
Nota: A variável tem que ter permissão de escrita.

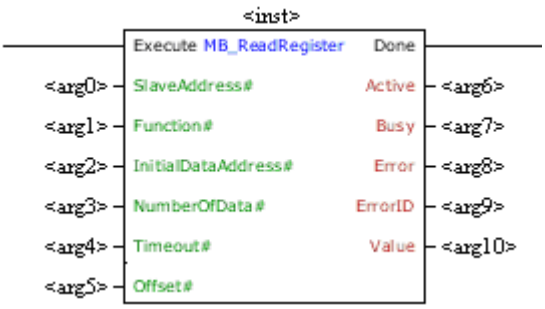
Descrição de Funcionamento

Bloco para leitura de bits. Permite fazer a leitura de até 128 bits em sequência do escravo destino, utilizando as funções 1 (Read Coils) e 2 (Read Discrete Inputs) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa, e os dados recebidos são copiados para "Value". Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

10.1.1.28.3 MB_ReadRegister

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Código da função de leitura: 3= "Read Holding Registers"; 4= "Read Input Registers".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Endereço do registrador inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Número de registradores lidos a partir do endereço inicial - 1 a 8.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.
Tipos de dados: BOOL
Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg6>: "Active" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco ativo, requisição de leitura enviada para o escravo e aguardando resposta.
Nota: A variável tem que ter permissão de escrita.

<arg7>: "Busy" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.
Nota: A variável tem que ter permissão de escrita.

<arg8>: "Error" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Erro na execução da requisição.
Nota: A variável tem que ter permissão de escrita.

<arg9>: "ErrorID" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".
Nota: A variável tem que ter permissão de escrita.

<arg10>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE[1 ... 16], SINT[1 ... 16], USINT[1 ... 16], WORD[1 ... 8], UINT [1 ... 8], INT[1 ... 8], DWORD[1 ... 4], UDINT[1 ... 4], DINT[1 ... 4] ou REAL[1 ... 4]
Descrição: Variável ou array onde serão salvos os dados lidos do escravo.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para leitura de registradores de 16 bits. Permite fazer a leitura de até 8 registradores em sequência do escravo destino, utilizando as funções 3 (Read Holding Registers) e 4 (Read Input Registers) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa, e os dados recebidos são copiados para "Value". Em caso de

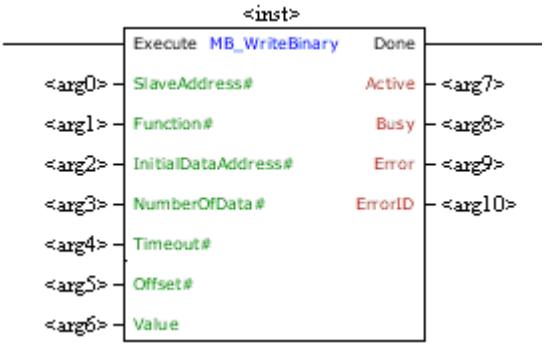
erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Nota

- O protocolo Modbus RTU, utilizando as funções 3 e 4, permite a leitura de registradores de 16 bits apenas; para leitura de dados com mais de 16 bits (um REAL, por exemplo), é possível fazer a leitura de múltiplos registradores, e salvar o valor em uma variável com tamanho maior que 16 bits, e
- É importante que a quantidade de registradores lidos seja compatível com o tamanho da variável ou do array onde os dados serão salvos.

10.1.1.28.4 MB_WriteBinary

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Código da função de escrita: 5= "Write Single Coil"; 15= "Write Multiple Coils".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Endereço do bit inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Número de bits escritos em sequência a partir do endereço inicial - 1 a 128.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD
Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.
Tipos de dados: BOOL
Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg6>: "Value" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BOOL[1 ... 128]
Descrição: Variável ou array com os dados que serão escritos no escravo.

<arg7>: "Active" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco ativo, requisição de escrita enviada para o escravo e aguardando resposta.
Nota: A variável tem que ter permissão de escrita.

<arg8>: "Busy" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.
Nota: A variável tem que ter permissão de escrita.

<arg9>: "Error" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Erro na execução da requisição.
Nota: A variável tem que ter permissão de escrita.

<arg10>: "ErrorID" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

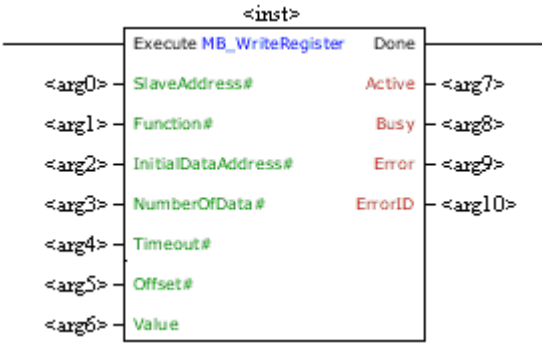
Bloco para escrita de bits. Permite fazer a escrita de até 128 bits utilizando as funções 5 (Write Single Coil) e 15 (Write Multiple Coils) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a

entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

10.1.1.28.5 MB_WriteRegister

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Código da função de escrita: 6= "Write Single Register"; 16= "Write Multiple Registers".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Endereço do registrador inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Número de registradores escritos a partir do endereço inicial - 1 a 8.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.
Tipos de dados: WORD
Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.
Tipos de dados: BOOL
Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg6>: "Value" - VAR_IN: inserir uma variável (tag).
Tipos de dados: BYTE[1 ... 16], USINT[1 ... 16], SINT[1 ... 16], WORD[1 ... 8], UINT [1 ... 8], INT[1 ... 8], DWORD[1 ... 4], UDINT[1 ... 4], DINT[1 ... 4] ou REAL[1 ... 4]
Descrição: Variável ou array com os dados que serão escritos no escravo.

<arg7>: "Active" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco ativo, requisição de escrita enviada para o escravo e aguardando resposta.
Nota: A variável tem que ter permissão de escrita.

<arg8>: "Busy" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.
Nota: A variável tem que ter permissão de escrita.

<arg9>: "Error" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Erro na execução da requisição.
Nota: A variável tem que ter permissão de escrita.

<arg10>: "ErrorID" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de registradores. Permite fazer a escrita de um ou mais registradores utilizando a função 6 (Write Holding Register) ou 16 (Write Multiple Registers) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

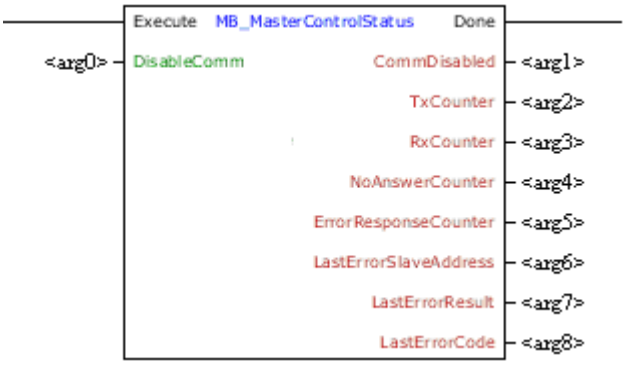
Nota

- O protocolo Modbus RTU, utilizando a função 16, permite a escrita de registradores de 16 bits apenas. Para escrita de dados com mais de 16 bits (um REAL, por exemplo), é possível fazer a escrita de múltiplos registradores, e utilizar como fonte

dos dados uma variável com tamanho maior que 16 bits,
- É importante que a quantidade de registradores escritos seja compatível com o tamanho da variável ou do array de onde os dados serão utilizados.

10.1.1.28.6 MB_MasterControlStatus

Figura



Configuração do Bloco

<arg0>: "DisableComm" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BOOL
Descrição: Permite desabilitar o mestre Modbus. Ao desabilitar o mestre, os contadores e marcadores de status do mestre Modbus RTU também são zerados: 0= "Mestre em execução"; 1= "Desabilita mestre".

<arg1>: "CommDisabled" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica se o mestre está ou não desabilitado. Pode ocorrer por solicitação do usuário ou caso a interface esteja programada para operar como escravo da rede: 0= "Mestre habilitado"; 1= "Mestre desabilitado".
Nota: A variável tem que ter permissão de escrita.

<arg2>: "TxCounter" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Contador de requisições enviadas pelo mestre da rede para os escravos. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.
Nota: A variável tem que ter permissão de escrita.

<arg3>: "RxCounter" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Contador de telegramas recebidos pelo mestre da rede. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.
Nota: A variável tem que ter permissão de escrita.

<arg4>: "NoAnswerCounter" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT

Descrição: Contador de requisições do mestre que não foram respondidas pelos escravos. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.
Nota: A variável tem que ter permissão de escrita.

<arg5>: "ErrorResponseCounter" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Contador de requisições do mestre e que os escravos responderam com alguma resposta de erro. O código do erro pode ser obtido no marcador que indica o código do último erro detectado. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.
Nota: A variável tem que ter permissão de escrita.

<arg6>: "LastErrorSlaveAddress" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Indica o endereço do escravo no qual foi detectado o último erro de comunicação. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 247.
Nota: A variável tem que ter permissão de escrita.

<arg7>: "LastErrorResult" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Indica o resultado da operação - timeout ou resposta de erro, conforme ERROR ID do bloco - para o escravo no qual foi detectado o último erro de comunicação. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado: 0= "Sem erro detectado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".
Nota: A variável tem que ter permissão de escrita.

<arg8>: "LastErrorCode" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BYTE ou USINT
Descrição: Indica o código do erro, no caso do mestre receber resposta de erro de algum escravo. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 255.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para controle e monitoração do mestre da rede Modbus RTU. Sempre que uma rede Modbus RTU for montada com o PLC300 como mestre da rede, recomenda-se utilizar este bloco para obter informações sobre o estado da comunicação.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Enquanto a entrada de habilitação "Execute" estiver ativa, os dados de entrada são utilizados e os dados de saída são atualizados. Caso a entrada seja zerada, os valores de entrada são desconsiderados e os argumentos de saída são zerados. A saída "Done" reflete o valor da entrada.

Nota

- Os dados acessados utilizando este bloco também estão disponíveis através de marcadores de sistema de leitura e escrita, conforme descrito no item "Marcadores

de Sistema do Modbus".

10.1.1.28.7 MB_SlaveStatus

Figura

Configuração do Bloco

<arg0>: "ErrorsToSetOffline#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Permite programar, para este bloco, a quantidade de erros de comunicação que o mestre deve identificar até que a comunicação com um escravo da rede seja considerada offline. É considerado erro de comunicação toda requisição (leitura ou escrita) que o mestre enviou para um escravo e não recebeu resposta, a resposta recebida possuía erro de CRC ou o escravo retornou resposta de erro - 1 a 99.

<arg1>: "AddressSlave1#" - VAR_IN: inserir uma constante.
<arg2>: "AddressSlave2#" - VAR_IN: inserir uma constante.
<arg3>: "AddressSlave2#" - VAR_IN: inserir uma constante.
<arg4>: "AddressSlave2#" - VAR_IN: inserir uma constante.
Tipos de dados: BYTE
Descrição: Permite programar o endereço de até 4 escravos, cuja quantidade de erros de comunicação serão monitorados para saber se estão online ou offline. Caso a quantidade de erros de comunicação em sequência, detectados nos blocos de leitura e escrita via Modbus, atinja o valor programado em "ErrorsToSetOffline", a saída respectiva é acionada. Caso deseje-se monitorar um número menor de escravos, pode-se deixar qualquer das entradas em zero: 0= "Ignora entrada"; 1 a 247.

<arg5>: "GeneralOffline#" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Se qualquer uma das saídas dos escravos indicados for acionada, esta saída também será acionada. Funciona como uma lógica OU entre as 4 saídas de indicação dos escravos - 0 a 65535.
Nota: A variável tem que ter permissão de escrita.

<arg6>: "Slave1Offline#" - VAR_OUT: inserir uma variável (tag).
<arg7>: "Slave2Offline#" - VAR_OUT: inserir uma variável (tag).
<arg8>: "Slave3Offline#" - VAR_OUT: inserir uma variável (tag).
<arg9>: "Slave4Offline#" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Saída acionada caso a quantidade de erros de comunicação em sequência

para os escravos indicados nas respectivas entradas atinja o valor programado em "ErrorsToSetOffline".
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para monitoração dos escravos da rede Modbus RTU. Deve ser utilizado caso seja desejado identificar problemas na comunicação do mestre com algum escravo da rede Modbus RTU.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Enquanto a entrada de habilitação "Execute" estiver ativa os dados de entrada são utilizados e os dados de saída são atualizados a cada execução do bloco. A saída "Done" reflete o valor da entrada.

Índice

- A -

ABS 97
ACOS 111
ADD 99
Adicionar/Remover Conexões 23
Alarmes 60
Alarmes Configuráveis 62
Alarmes Internos 63
ALOG10 106
Ambiente 15
AND 88
Antes de Instalar 11
ASHL 92
ASHR 93
ASIN 110
ATAN 112
ATAN2 113
Atualizando Driver USB 27
Aviso 10

- B -

BCD_TO_WORD 137
Bem Vindo 8
BOOL_TO_BYTE 126
BOOL_TO_DWORD 132
BOOL_TO_REAL 134
BOOL_TO_WORD 129
BYTE_TO_BOOL 123
BYTE_TO_DWORD 132
BYTE_TO_REAL 135
BYTE_TO_WORD 129

- C -

Cabos 24
CANopen 173
CO_MasterControlStatus 179
CO_SDORead 175
CO_SDOWrite 177
CO_SlaveStatus 181
COMPEQ 117
COMPGE 118
COMPGT 119
COMPLE 120
COMPLT 121
COMPNE 122
Componente Bargraph 71
Componente Message 72
Componente Numeric Input 69
Componente Numeric Output 70
Componete Text 68
Configuração 23
Configuração das Telas 53
Configuração de Tarefas 46
Contato 9
Conversor USB/Serial 25
COS 109
CTD 151
CTU 152
CTUD 153

- D -

Dados das Variáveis 33
DEG_TO_RAD 139
DEMUX 147
Descrição 75
Desinstalando 14
Desinstalando Driver USB 27
DIRECTCOIL 81
DIV 102
Download 9
DWORD_TO_BOOL 124
DWORD_TO_BYTE 127
DWORD_TO_REAL 136
DWORD_TO_WORD 130

- E -

Edição de Variáveis na Rung 43
Editando Título e Comentário da Rung 36
Editando uma Rung 35
Editor de Variáveis/Ladder 31
Editor Ladder/Rungs 34
Editor/Lista de Variáveis 31

Equipamentos Suportados 12
Executando 23
EXP 104

- G -

Gerenciador de Comunicação 14
Gerenciamento de Telas 55
Glossário 76
Guia de Inicialização Rápida 15

- I -

Identificação 8
ILOAD 142
ILOADBOOL 141
IMMEDIATE_INPUT 167
IMMEDIATE_OUTPUT 168
IMMEDIATECOIL 84
Informações Legais 9
Inserindo Elementos em Paralelo 43
Inserindo Elementos em Série 42
Inserindo Elementos na Rung 37
Instalando 13
Instalando Driver USB 26
INTIME 169
INWEEKDAY 171
ISTORE 144
ISTOREBOOL 143

- J -

Java Runtime Environment 12

- L -

LN 104
LOG10 105
LOWPASS 162

- M -

Manipulação de Bits 56
MAX 113
MB_MasterControlStatus 191
MB_ReadBinary 183
MB_ReadRegister 185
MB_SlaveStatus 193
MB_WriteBinary 187
MB_WriteRegister 189
MIN 114
MOD 103
Modbus 182
Monitoração Online Ladder 73
MUL 101
MUX 145

- N -

Navegação pelo Mouse na Rung 40
Navegação pelo Teclado na Rung 38
NCCONTACT 78
NEG 98
NEGATEDCOIL 81
NOCONTACT 78
NOT 88
Novidades 8
NTSCONTACT 80

- O -

OR 89

- P -

PID 159
POW 107
Programação das Telas 64
Programação Ladder 29
PTSCONTACT 79
PWM 166

- R -

RAD_TO_DEG 138
READENC 163
READENC2 164
READENC3 165
REAL_TO_BOOL 125
REAL_TO_BYTE 128
REAL_TO_DWORD 134
REAL_TO_WORD 131

Requisitos Mínimos 11
RESETBIT 85
RESETCOIL 83
ROL 96
ROR 97

- S -

SAT 115
SEL 149
Senhas 58
SETBIT 86
SETCOIL 82
SETSCREEN 172
SHL 94
SHR 95
SIN 108
SQRT 108
STORE 140
SUB 100
Suporte Técnico 9
SWAP 150

- T -

TAN 110
Teclas de Função 55
TESTBIT 87
Tipos de Dados 75
TOF 156
TOGGLECOIL 83
TON 155
TP 157

- V -

Verificando a Instalação do Driver USB 28
Visão Geral 8

- W -

WORD_TO_BCD 138
WORD_TO_BOOL 123
WORD_TO_BYTE 126
WORD_TO_DWORD 133
WORD_TO_REAL 136

- X -

XNOR 91
XOR 90